BROADCOM®
SOFTWARE

Layer7 API Management SaaS

Table of Contents

Change Log **6**

5.1.2 - 2022-10-25 (S), 2022-11-01 (P) **6**

5.1.1 - 2022-06-28 (S), 2022-07-05 (P) 11

5.1 - 2022-03-22 (S), 2022-03-29 (P) 12

5.0.3- 2021-09-14 (S), 2021-09-21 (P) 14

5.0.2- 2021-06-29 (S), 2021-07-06 (P) 16

5.0.1 - 2021-02-16 (S), 2021-02-23 (P) 18

5.0.0 - 2020-10-20 (S), 2020-10-27 (P) 20

4.5.x **23**

4.5.5 - 2020-09-15 (S), 2020-09-22 (P) 23

4.5.4 - 2020-08-04 (S), 2020-08-11 (P) 25

4.5.3 - 2020-07-07 (S), 2020-07-14 (P) 25

4.5.2 - 2020-06-09 (S), 2020-06-16 (P) 27

4.5.1 - 2020-05-01 (S), 2020-05-20 (P) 28

4.5.0 - 2020-05-02 (P) 29

4.4 and Earlier **30**

Known Issues: API Management SaaS **41**

Resolved Issues: API Management SaaS **45**

Compatibility Matrix **53**

Set Up and Maintenance **55**

Set Up API Management SaaS **55**

Integrate On-Premise API Proxies **56**

Manage URLs for API Proxy Enrollment **61**

Configure Authentication Schemes **62**

Configure Microsoft Active Directory 63

Configure Lightweight Directory Access Protocol 66

Configure SSO for Local SaaS Environments 69

Configure SAML Single Sign-On 79

FAQ 84

Set Default Authentication Scheme 88

Manage Password Policy 88

Map IdP Users to Multiple Organizations 89

Configure User Registration **92**

Configure Request Workflow **94**

Application Requests **96**

Manage Requests from Developers **97**

Configure Mail Server at Tenant Level 98
Customize API Portal Pages 100
Customize Page Appearance 101
Customize Core Pages 103
Manage Custom Fields 108
Set Up Custom Domain Names 113
Enable Google Analytics Tracking 118
Configure Security 118
Enable Hashed Client Secret 119
Audit Logs 121
Update the Gateway with New Portal Certificates 124
Re-Import Expired APIM Ingress Certificate 126
Update Portal Integration Software 126
User Types, Roles and Permissions 128
Manage Users 132
Manage Organizations 135
API Portal Dashboard 143
Manage 147
Manage APIs 152
Create and Set Permissions for APIs 155
View the APIs and Applications on the Developer Console 160
Edit and Delete APIs 161
Manage API Tags 165
Publish APIs with Additional Configurations with the API Portal 166
Publish APIs with the API Proxy and Policy Manager 172
Test and Explore APIs 175
Explore APIs 176
Manage API Deployments 178
API Deployment Types 178
Deploy APIs 181
Deploy APIs to Proxies using PAPI 182
Troubleshoot API Deployments 225
Manage API Monitoring Tests 227
Manage API Documents 230
Manage API Lifecycles and States 231
Manage Applications 233
Manage API Keys 237
Manage API Key Deployments to Proxies 240
Deploy to Proxies using Portal 241

Deploy to Proxies using PAPI 243
Troubleshoot API Key Deployments 246
Work with Applications 248
Manage API EULAs 251
Manage Proxies 252
Manage API Usage 258
Manage Rate Limits and Quotas 260
Manage API Plans 265
Working with API Plans 266
View and Choose API Plans 268
Manage API Groups 269
Manage Policy Templates 271
Add Policy Templates using Policy Manager 272
Manage Policy with Gateway Bundles 273
Upload and Deploy Gateway Bundles 276
Associate Policy Templates to API 279
Monitor 281
Traffic, Latency and Errors Report 281
Quota Consumption Report 283
Custom Report 285
Hardware Optimizer 290
Portal APIs 292
Portal API (PAPI) 292
PAPI Swagger File 5.1.2 296
Portal Metrics API 296
Metrics Query API 296
Login API 313
Authorization API 315
API Hub 317
Getting Started with API Hub 318
Configure the Standard API Hub at Runtime 322
Customize and Extend the Standard API Hub 322
Access API Hub 325
Manage the Content in API Hub 326
View APIs using API Hub 327
Manage Applications using API Hub 328
Manage Wiki Documents in API Hub 331
Portal Training Videos on IMS Software Academy 334
Third-Party Software Acknowledgments 336

Product Accessibility Features 337
Documentation Legal Notice 339

Change Log

This is the change log for Layer7 API Management SaaS.

For a full list of known and fixed issues, see Known Issues and Resolved Issues.

> **IMPORTANT**
>
> **About Staging and Production Environments**
>
> Each API Management SaaS upgrade is available to test in a staging environment before it is pushed to production. Refer to the dates in this change log for:
>
> - The date that the upgrade is available in the staging environment (S), and
> - The date that the upgrade is pushed to production (P).

5.1.2 - 2022-10-25 (S), 2022-11-01 (P)

Summarizes the features that have been added to or changed for the current release of the API Developer Portal.

> **IMPORTANT**
>
> **About Staging and Production Environments**
>
> Each API Management SaaS upgrade is available to test in a staging environment before it is pushed to production. Refer to the dates in this change log for:
>
> - The date that the upgrade is available in the staging environment (S), and
> - The date that the upgrade is pushed to production (P).

Contents:

- Added
- Changed
- Experimental
- API Developer Portal Release and Support Lifecycle Dates

Added

Support for Custom Pages Restored

Previously in version 5.0.2 of the API Portal (SaaS), Layer7 announced the deprecation of Custom Pages. For the release of version 5.1.2, this feature has been restored after receiving feedback provided by the Layer7 community.

In effect, the PAPI endpoint for custom page resources (/custom/1.0/pages) has also been restored.

API Portal users may continue to use the API Hub for Portal customization work as required. To use this feature, you must submit a request to our SaaS support team for activation.

> **ATTENTION**
>
> **Custom Pages for Portal Versions 5.1.2 or Newer: Inline JavaScript Prohibited**
>
> Content-Security-Policy is a HTTP response header that web browsers use to enhance the security of a document or web page. To comply with current updates in the Content-Security-Policy header, inline JavaScript is now blocked from being executed in a web browser. If you have any existing custom pages or plan to have custom pages that contain inline JavaScript, you must extract those JavaScript elements into a separate script and reference them as an src include instead.

Rate Limits and Quotas Enhancements

> **NOTE**
> In order to make use of the latest enhancements for Rate Limits and Quotas in the Portal, ensure that version 2.0.1 of the rate limit and quota policy template, l7.apim.system - Rate & Quota Policy Template - 2.0.1, has been applied and deployed to enrolled proxies.

New Configurable Default Value

Portal administrators and API owners can configure a global default value for 'API per Organization' rate limit and quota, eliminating the requirement to separately configure one for each individual API. To learn more, see Manage Rate Limits and Quotas.

Hourly Quota Consumption Report by API and API per Organization

The API Developer Portal 5.1.2 and later includes an hourly quota consumption chart. In addition, Monthly, Daily, and Hourly quota consumption reports are supported at the following levels:

- API
- Organization API per
- Organization

For more information about the Hourly Quota Consumption Report, seeQuota Consumption Report.

Fully Customized Email Notification Templates

You can now retrieve and update customized Portal email notification templates via the latest Portal API (PAPI). The list below shows some sample calls you can make for the new PAPI endpoints that support this new feature:

GET /tenant-admin/1.0/email-templates	Retrieves full list of email templates.
GET /tenant-admin/1.0/email-templates/{uuid}	Retrieves template type, subject, and HTML body of the email template identified by its uuid.
PUT /tenant-admin/1.0/email-templates/{uuid}	Updates email template with the given uuid.

The following shows the following editable fields for a PUT update on an email template:

```
{
    "uuid": "caaaa-8a68-4dsb-4b34-846l-006348453a" ,
    "type": "EMAIL_FOR_APPLICATION_REQUEST" ,
    "subject": "Application Delete Request for ((Portal_NAME)) has been Approved" ,
    "body": "<!DOCTYPE html><html><body><h1 style=\"background-color: red; \"
            >Notification!</h1> <p>Hello:</p>....</body></html>
{
```

> **NOTE**
> - The "body" field accepts HTML in a single continuous line.
> - The insertion of scripts is prohibited in the email template and shall return an error indicating a potential malicious HTML code.

See the latest PAPI 5.1.2 swagger file to learn more about the new customized email template endpoints.

Public API Information Available without Authentication

The Portal now allows the bypassing of client authentication when fetching API details and various asset types associated with public APIs. This eliminates the need to manage users for public API information that's free to share.

The existing public API catalog endpoint can now return public APIs with an optional status (ENABLED or DEPRECATED), spec file (i.e., swagger file), or tag filter applied:

- GET /api-management/1.0/api-catalogs**?portalStatus=DEPRECATED**
- GET /api-management/1.0/api-catalogs**?hasSpecFile=true**
- GET /api-management/1.0/api-catalogs**?tags=sampletag**

The following are sample calls you can make on various asset types associated with public APIs:

- GET /api-management/1.0/api-catalogs/**{apiUuid}/assets**
- GET /api-management/1.0/api-catalogs/**{apiUuid}/assets/swagger**
- GET /api-management/1.0/api-catalogs/**{apiUuid}/assets/wsdl**
- GET /api-management/1.0/api-catalogs/**{apiUuid}/assets/{assetUuid}/file**

See the latest PAPI 5.1.2 swagger file to learn more about the new api-catalogs endpoints.

NOTE

You may activate this feature with following feature flag:

PUBLIC_API_CATALOG_ENABLED

For example, using the /Settings PAPI endpoint, your PUT payload to activate the feature should resemble the following:

```
{
  "Name": "PUBLIC_API_CATALOG_ENABLED",
  "Uuid": "00000000-0000-0000-0000-000000000000",
  "Value": "true"
}
```

Changed

API Details Tab in Organization Details

In addition to being able to view the list of APIs along with their accessibility status at an organization level, administrators and API owners can now add or remove access to APIs, and assign a rate limit and quota at the ‘API per Organization’ assignment level from the Organizations Details page.

Administrators may continue to assign rate limits and quotas at the organization assignment level from the Organization Details > Overview tab.

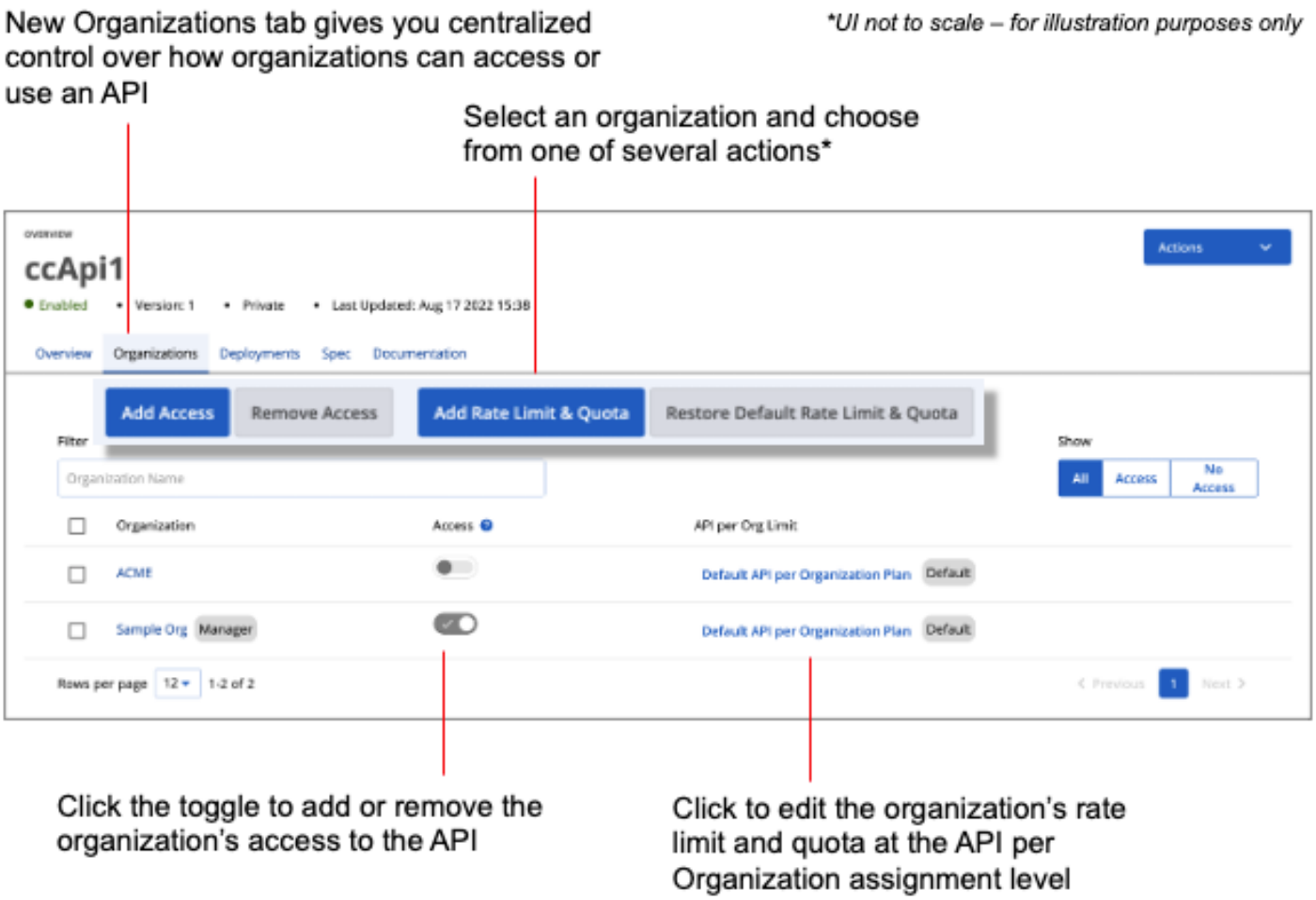
To learn how to make full use of the new APIs tab , see Manage Organizations.

Configuration of API Access Permissions Moved to API Details

Previously, the setting up of API access permissions (previously known as “visibility”) for organizations was done via the Add/Edit API wizard. For the release of Portal version 5.1.2, this configuration has been moved to the new ‘Organizations’ tab in the API Details page. The ‘Organizations’ tab provides the following information:

- Lists all organizations that currently have access or don’t have access to the API
- The currently assigned rate limit and quota for each organization at the ‘API per Organization’ level (each organization can be assigned a custom API per organization rate limit and quota OR the default one)

See Edit and Delete APIs to learn more about the Organizations tab and how you can use it to manage private API access for organizations and rate limits and quotas.



Enhanced Documentation Tab in API Details

The Documentation tab has been streamlined to make it easier for document owners to author and organize their documents.

Experimental

Enhancements for Monitoring Expired and Expiring Trusted Certificates (Experimental Feature)

The enhancement is present in both the Portal UI and a new PAPI endpoint, and is categorized as an 'Experimental Feature' for this release.

Prerequisite: The Graphman assertion (Experimental Feature) must be installed in order for this feature to operate.

Layer7 Broadcom does not recommend the use of this feature in a production environment. See Progressive Delivery of APIM Features to learn about the Experimental feature category and others.

Alerts for Expiring and Expired Certificates (Experimental Feature)

The count of trusted certificates that are expiring or have expired is now indicated in the Proxy Errors list. For more information, see API Portal Dashboard.

View Summary of Expired and Expiring Trusted Certificates in Tenant Gateways (Experimental Feature)

The following GET call and PAPI endpoint returns a list of proxies (identified by their Uuid and Name) with a certificate type of “EXPIRED” and/or “EXPIRING” and a count of expired or expiring certificates:

GET /gateway-management/**0.1/status/trusted-certs**

You can also append the following optional query parameters:

GET /gateway-management/0.1/status/trusted-certs?expiringInDays={#ofdays}	Returns a list of proxies with expiring or expired certificates that fall within the specified number of days remaining till expiry. If left unspecified, the default value is 60 days, or return proxies with certificates that are expired and/or expiring within 60 days.
GET /gateway-management/0.1/status/trusted-certs?proxyUuid={proxyUuid}	Returns a proxy of a specific Uuid and its expiring and/or expired certificates status.

NOTE

You may activate this experimental feature with the following feature flag:

FEATURE_FLAG_L7IM

API Developer Portal Release and Support Lifecycle Dates

For more information on product release support lifecycle dates and upcoming End-Of-Service dates, see the API Developer Portal Release and Support Lifecycle Dates page.

5.1.1 - 2022-06-28 (S), 2022-07-05 (P)

IMPORTANT

About Staging and Production Environments

Each API Management SaaS upgrade is available to test in a staging environment before it is pushed to production. Refer to the dates in this change log for:

- The date that the upgrade is available in the staging environment (S), and
- The date that the upgrade is pushed to production (P).

Added

- **Rate Limits and Quotas Enhancement: Introducing the 'API per Organization' Assignment Level**
  In addition to the existing rate limit and quota assignment levels of 'API' and 'Organization', the API Portal now allows administrators to assign at the *API per organization* assignment level to allow for increased granular API usage control. More importantly, this new assignment level allows you to constrain API usage that is specific to an organization. Previously, usage constraints were either imposed on:
  - An organization, on which the constraint applies to all API usage by that organization (the organization assignment level) or
  - The API itself, regardless of who the API consumer is (the API assignment level).

  The third assignment level, or the *API per organization* assignment level, fills in the gap between the organization and API assignment levels. Organizations are no longer limited to a single rate limit and quota; different rate limits and quotas can now be assigned to a specific API per organization. To learn more about this new assignment level and how rate limits and quotas are handled, see Manage Rate Limits and Quotas.

IMPORTANT
In order to make use of the new API per organization assignment level, ensure that version 2.0 of the rate limit and quota policy template, *l7.apim.system - Rate & Quota Policy Template - 2.0*, has been applied and deployed to enrolled proxies.

- **View List of APIs at an Organization Level**
  Portal Administrators and API Owners can now view the list of APIs along with their accessibility status at an organization level. You can also add or remove access to APIs from the Organizations page in Portal UI. For more information about how to view the list of APIs at an organization level, see Manage Organizations.

  New Portal APIs (PAPI)
  Introduced the following Portal APIs:

  - GET: **/organizations/{org-uuid}/apis/**
    Lists all the Private APIs that are accessible and that can be updated by the Organization.
  - PUT: **/organizations/{org-uuid}/apis/**
    Updates the list of APIs that are allowed to be accessible for the Organization.
  - PATCH: **/organizations/{org-uuid}/apis?action={add/remove}**
    Adds or removes based on the action parameter, the list of APIs that are allowed to be accessible to the Organization.

  For more information about these PAPIs, see the PAPI swagger file.
- **Application Request Workflow Includes Delete Application**
  Org Admins and Developers can now delete applications after acquiring approval from the Portal Administrator. For more information about how to enable or disable the delete application request workflow, see Configure Request Workflow.

Changed

- **Changes in API Visibility Options for the Add/Edit API Wizard**
  The following sections of the Add/Edit API Wizard have been modified:

  - The option to set an API public or private has been moved to the Details section of the wizard.
  - The Organization Access section has been created to assist publishers in granting API access to specific organization(s) and selecting an appropriate rate limit and quota via the new 'API per organization' assignment level.
- **Improved System Architecture for Proxy Enrollments**
  The system architecture behind the enrollment process for Gateways in the Portal has been improved to simplify installation and maintenance.
- **Limits and Quotas Caching Mechanism Optimized for Enrolled Proxies**
  Cache optimization has improved the speed of changes to API rate limits and quotas configuration changes, as well as the speed of rate limits and quotas searches in the API Portal.

API Developer Portal Release and Support Lifecycle Dates

For more information on product release support lifecycle dates and upcoming End-Of-Service dates, see the API Developer Portal Release and Support Lifecycle Dates page.

5.1 - 2022-03-22 (S), 2022-03-29 (P)

IMPORTANT

About Staging and Production Environments

Each API Management SaaS upgrade is available to test in a staging environment before it is pushed to production. Refer to the dates in this change log for:

- The date that the upgrade is available in the staging environment (S), and
- The date that the upgrade is pushed to production (P).

Added

- **Rate Limits and Quotas for API Management**
  Portal Admins can now define 'Rate Limits and Quotas' to manage API usage at two different assignment levels: at the Organization level OR the API level. The 'Account Plans' configuration to manage rate limit and throughput quotas for Organizations is now incorporated into 'Rate Limits and Quotas' with the organization level. At the API level, API publishers want to ensure that they can provide the protection based on the load that can be handled by the API backend services.
  This new feature is intended to streamline Portal's rate limiting capability across organizations and APIs. Once API assignment Rate Limits and Quotas are defined, API publishers can select the new Rate Limit and Quota system policy template during API publishing, apply the pre-configured Rate Limit and Quota configurations to the API.
  To learn how Rate Limits and Quotas can work together with API Plans, see Manage API Usage.
  To learn more about the new Manage Rate Limits and Quotas function, see Manage Rate Limits and Quotas.
- **Enhanced Synchronization for Account Plans (Organization-level Rate Limits and Quotas)**
  The mechanism used to synchronize account plans (i.e., organization-level rate limits and quotas) between the Portal and API Gateway has been enhanced to improve reliability of synchronization operations.

  > **IMPORTANT**
  > To start using the enhanced synchronization method, ensure you have installed the latest update of the Portal Integration Bundle on the API proxy. For the latest version of the bundle supported by the Portal, see Compatibility Matrix.

- **Improved Security for User Management: Email Address Changes**
  In order to improve access security for user management, Portal Administrators must now re-authenticate before they can change a API Portal user's email address. Only Portal Administrators can update the email address in the My Profile page and API Hub. All other users such as API Owners, Org users, and developers cannot update their own email addresses in the Portal UI and API Hub.
  For information, see Manage Users and API Hub.
- **Multiple Critical Vulnerabilities Addressed**
  Several Portal artifacts and third-party libraries were updated in this Portal release to provide additional security and performance updates to Portal components.

Changed

- **Manage API Plans - Visibility Governance Changes**
  When creating or updating API plansand assigning organizations to an API plan, the selection of organizations is not limited to only those that have permissions to the APIs within the plan. You can assign any organization, regardless of their API visibility permissions, to the API Plan. The ability to apply this API plan during application creation shall be governed by the organization's visibility to the API.
- **Retain Target Page After SAML SSO Authentication**
  When you access a bookmark page and log in into API Developer Portal, you will now be redirected to the same bookmark page instead of the API Portal Dashboard or the admin page.
  If you are using the SAML SSO that has been deprecated leveraging the API Gateway and this functionality is not available after you upgrade API Developer Portal, ensure that the **API Portal SSO** policy is updated in API Gateway as explained in SAML FAQ.
- **Solr Search Removal**

Solr as an API search service has been removed in API Portal. Search functionality has been updated to use PAPI to search for APIs and Applications.

- **Runscope Integration Removal**
  The Runscope Integration has been removed from API Portal. Runscope is part of the BlazeMeter Continuous Testing platform. Users can still use Runscope independently to monitor APIs published in the API Portal.

Deprecated Features

Deprecated APIs

The following API has been deprecated as part of an ongoing effort to streamline API Portal's system architecture.

- Account Plans

Account Plans

The /AccountPlans endpoint has been deprecated and replaced by the /api-management/1.0/rate-quotas endpoint. Use the assignmentLevel filter to filter by Organization to return rate limit and quotas (previously known as account plans).

See the latest PAPI 5.1 swagger file to learn more about the new endpoint(s).

> **IMPORTANT**
> Service for deprecated endpoint(s) have been moved to the deprecated section of the PAPI swagger file and will be removed from the next Portal release. Broadcom recommends users to transition over to the new replacement endpoints as soon as possible.

Discontinued APIs

The previously deprecated /search/query and /search/autosuggest APIs have now been discontinued (see deprecated APIs section of the PAPI Swagger File 5.0.2 for a detailed description of this resource).

Moving forward, users are encouraged to use the GET /api-management/1.0/apis and GET /api-management/1.0/applications APIs instead. For more details, see the PAPI Swagger File 5.1.

API Developer Portal Release and Support Lifecycle Dates

For more information on product release support lifecycle dates and upcoming End-Of-Service dates, see the API Developer Portal Release and Support Lifecycle Dates page.

5.0.3- 2021-09-14 (S), 2021-09-21 (P)

> **IMPORTANT**
>
> **About Staging and Production Environments**
>
> Each API Management SaaS upgrade is available to test in a staging environment before it is pushed to production. Refer to the dates in this change log for:
>
> - The date that the upgrade is available in the staging environment (S), and
> - The date that the upgrade is pushed to production (P).

Added

- **Portal Dashboard: New Proxy Errors List**
  In order to act quickly on API, API key deployment, or Proxy connection issues, Portal administrators must rely on the Portal to inform them of these issues as they arise. In an ongoing effort to improve the manageability of Proxies enrolled with API Portal, Portal administrators can now view Proxy connection issues and any API or API key deployment issues in the new Proxy Errors list from the API Portal dashboard. See API Portal Dashboard to learn more.

IMPORTANT

To be able to use the new Proxy Errors list in the API Portal Dashboard, ensure that you have installed the latest update of the Portal Integration Bundle on the API proxy. For the latest version of the bundle supported by the Portal, see Compatibility Matrix.

- **Organization Tagging**
  Recall that organizations are a way to group and manage related developers. In addition to viewing organizations and assigning developers to them, Portal administrators can add and edit organizations in API Management SaaS. The Portal now supports tagging for organizations to help administrators group related organizations by way of tagging. See Manage Organizations to learn more.
  You can also manage organization tags using the new `/tenant-admin/1.0/tags` endpoint in the Portal API (PAPI). See the latest PAPI 5.0.3 swagger file to learn how you can use this endpoint to perform operations such as retrieving and creating organization tags, and updating tag-organization associations.
- **Support for EULA Updates**
  End-User License Agreements (EULAs) are a requirement for all published APIs. Portal Admins and API Owners can manage EULAs in the Portal, which now includes the ability to edit EULA content that is currently referenced by published APIs.
- **New Endpoints in Portal API (PAPI)**
  Portal 5.0.3 introduces a new set of endpoints labeled as `api-management/1.0/tags` for managing API tags and `tenant-admin/1.0/tags` for managing Organization tags. You can access these endpoints only through PAPI. The `api-management/1.0/tags` endpoint replaces the now deprecated `/tags` endpoint.
  - **Support for Filtering and Deleting Tags**
    Portal Administrators can now filter tags using the **inUse=True/False** filter and delete tags that are not associated with any API or organization using the new endpoint in the Portal API (PAPI).
  - **Support for Renaming Tags**
    Portal Administrators can now rename tags after it is created using the new endpoints.

  Another new endpoint, `api-management/1.0/eulas` has also been introduced for this release, allowing Portal users to perform tasks such as listing EULAs that can be applied to APIs, creating an API EULA, or updating/deleting an existing API EULAs. This new endpoint effectively replaces the now deprecated `/ApiEulas` endpoint.
  For more information, see Manage API Tags. For information about the new endpoints, see the latest PAPI 5.0.3 swagger file.

Changed

- **Enhanced Synchronization for API Plans**
  The mechanism used to synchronize API plans between the Portal and API Gateway has been enhanced to improve reliability of synchronization operations.

  IMPORTANT
  To start using the enhanced synchronization method for API plans, ensure you have installed the latest update of the Portal Integration Bundle on the API proxy. For the latest version of the bundle supported by the Portal, see Compatibility Matrix.
- **Spring Boot Library Upgrades**
  The Spring Boot Libraries used in API Portal have been updated to provide additional security and performance updates to Portal components.

Deprecated Features

Deprecated APIs

The following APIs have been deprecated as part of an ongoing effort to streamline API Portal's system architecture:

- API EULAs
- Tags

API EULAs

The `/ApiEulas` endpoint has been deprecated and replaced by the new `api-management/1.0/eulas` endpoint.

Tags

The `/tags` endpoint has been deprecated and replaced by the `api-management/1.0/tags` endpoint.

See the latest PAPI 5.0.3 swagger file to learn more about the new endpoints.

> **IMPORTANT**
> Service for the deprecated endpoints have been moved to the deprecated section of the PAPI swagger file and will be removed from the next Portal release. Broadcom recommends users to transition over to the new replacement endpoints as soon as possible.

Runscope Integration Deprecation

The integration between API Portal and Runscope is deprecated and will be removed from the next release of the Portal. API monitoring tests that you have created from API Portal will continue to run independently in Runscope.

API Developer Portal Release and Support Lifecycle Dates

For more information on product release support lifecycle dates and upcoming End-Of-Service dates, see the API Developer Portal Release and Support Lifecycle Dates page.

5.0.2- 2021-06-29 (S), 2021-07-06 (P)

> **IMPORTANT**
>
> **About Staging and Production Environments**
>
> Each API Management SaaS upgrade is available to test in a staging environment before it is pushed to production. Refer to the dates in this change log for:
>
> - The date that the upgrade is available in the staging environment (S), and
> - The date that the upgrade is pushed to production (P).

Added

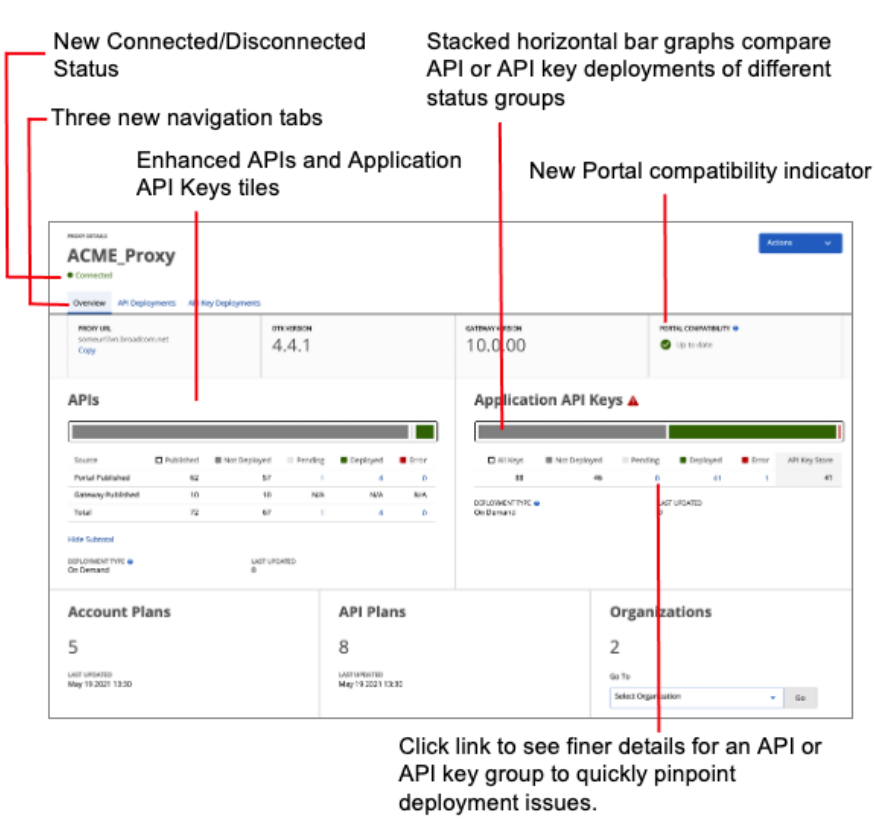
- **API Hub: View Multiple API Keys in API Applications**
  Application developers can now view multiple keys associated with an application from the API Hub.
  To download the latest updates on API Hub, see the CA APIM GitHub page.

Changed

- **Enhanced Synchronization for Automatic API and API Key Deployments**
  The mechanism used to synchronize automatic API and API key deployments between the Portal and API Gateway has been enhanced to improve reliability and scalability for high-volume API synchronization operations. To learn more about this enhancement and its other benefits, see Manage Proxies.

  > **IMPORTANT**
  > To start using the enhanced synchronization method for automatic API and API key deployments on your API proxy, you must install the latest update of the Portal Integration Bundle on the API proxy.
- **Improved Proxy Details Page**
  The Proxy Details page shows Portal administrators how API Gateways are currently enrolled and configured to work with the API Management SaaS and has been improved in several ways. The page has been redesigned to provide a

more practical and informational dashboard experience for Portal administrators who want to quickly understand the status of their proxies and effectively gain insights from all deployment activities managed by the API Management SaaS. The following is a highlight summary of those improvements. For a detailed summary of the new Proxy Details page, see Manage Proxies.



- **New Connected/Disconnected Status**: Take out the guesswork by determining immediately whether your API proxy is experiencing any connection issues with the Developer Portal.
- **Three New Navigation Tabs**: Users can cycle through three unique views of API proxy details, including a high-level overview (as shown), API Deployments, and API Key Deployments.
- **New Compatibility Indicator**: Quickly determine if your proxy has the latest Portal Integration Bundle installed to leverage the latest integration capabilities between the Portal and Gateway.

- **Enhanced APIs and Application API Key Tiles**: These tiles now show groupings of API and API Key counts by source and deployment status. You can also click individual counts to see finer details for each unique grouping in their respective API Deployments or API Key Deployments tabs and quickly identify deployment problems if they arise.
- **New Stacked Horizontal Bar Graph**: Lets administrators easily compare API or API key deployments statuses by colour segmentation.

- **Spring Boot Library Upgrades**
  The Spring Boot Libraries used in API Portal have been updated to provide additional security and performance updates to Portal components.

Fixed

- **Multiple Defect Fixes Completed for /v2/users API**:
  The following fixes were made to the /v2/users API endpoint:

- HATEOAS links have been changed to relative links to align with the behaviour of other APIs (they were incorrectly returned as absolute URLs prior to the fix)
- Validation of invalid values inserted in the Limit and Offset fields now result in a '404 (NOT_FOUND)' response instead of '400 (BAD_REQUEST)'
- UUID format validation check is now enforced as a priority for API calls
- Minor endpoint permissions issues have been resolved

Deprecated and Discontinued Features

Deprecated APIs

The following APIs have been deprecated as part of an ongoing effort to streamline API Portal's system architecture:

- Custom Pages
- Search

Custom Pages

The custom page resources assist organizations in customizing their developer console. They include:

- `GET /custom/1.0/pages` to return all custom pages.
- `POST /custom/1.0/pages/assign` to update custom page assignments.

Moving forward, users are encouraged to use the API Hub for all developer console customization tasks as an alternative.

Search

Search resources are used to search for APIs and applications in the Portal dashboard. They include:

- `GET /search/query` to search for Portal entities by keyword.
- `GET /search/autosuggest` to search for entities by autosuggested terms.

Alternatively, users are asked to use the following APIs instead to perform their API or application searches:

- `GET /api-management/1.0/apis`
- `GET /api-management/1.0/applications`

Discontinued APIs

The previously deprecated `/Users` APIs have now been discontinued (see deprecated APIs section of the PAPI Swagger File 5.0 for a detailed description of this resource).

Moving forward, users are encouraged to use the `/v2/users` API instead. For more details, see PAPI Swagger File 5.0.2.

API Developer Portal Release and Support Lifecycle Dates

For more information on product release support lifecycle dates and upcoming End-Of-Service dates, see the API Developer Portal Release and Support Lifecycle Dates page.

5.0.1 - 2021-02-16 (S), 2021-02-23 (P)

IMPORTANT

About Staging and Production Environments

Each API Management SaaS upgrade is available to test in a staging environment before it is pushed to production. Refer to the dates in this change log for:

- The date that the upgrade is available in the staging environment (S), and
- The date that the upgrade is pushed to production (P).

IMPORTANT
Note for customers using API Gateway 9.2 or 10 CR1 and higher

For customers using the API Portal with the Layer7 API Gateway version 9.2 (any CR) or version 10 CR1 and higher, review the following information:

- **API Gateway 9.2 (any CR):** There is limited compatibility support when using API Portal 5.0 with API Gateway version 9.2. Do NOT perform an update of the integration bundle as part of your upgrade; otherwise, a number of new API Portal features will not be available. We recommend upgrading your API Gateway to a version higher than 9.2 (i.e., version 9.3 or higher) . See Compatibility Matrix for more information.
- **API Gateway 10 CR1 and higher:** You MUST update the integration bundle and replace the PortalUpgradeAssertion file as part of your upgrade. See Update the Integration Software on the API Proxy for more information.

IMPORTANT

Note for customers using API Gateway 10 CR1

For customers using the API Portal with the Layer7 API Gateway, a sync issue exists that renders some Portal-published APIs incompatible with API Gateway version 10 CR1. As a temporary workaround, we recommend using other versions of the API Gateway (9.x or 10.0 base) while a fix is being worked on.

Added

- **Introducing Organization Type for Organizations:**
  Portal Admins can now define an organization type, **Publisher** or **Consumer**, for each organization to differentiate between organizations that can manage and publish APIs and organizations that can only consume APIs. This provides security enforcement for organizations and organization-based users that are defined in API Portal.
  A **Publisher** organization can include users such as Org Publishers, Org Admin, and Developers who can manage and consume APIs. A **Consumer** organization can include only Org Admin and Developer users who can only consume APIs. An Org Publisher role cannot be created or updated in a Consumer organization.
  During API Portal upgrade, the organization type is automatically assigned based on the following conditions:
  - An organization that has Org Publisher role users, or manages APIs, or it is associated with any proxy is assigned the **Publisher** organization type.
  - An organization that does not have any of the above associations is assigned to the **Consumer** organization type.

  For information about the Organization Type, see Manage Organizations.
- **New Okta Single Sign-on for API Hub:**
  API Management SaaS users can now log in into API Hub using Okta single sign-on. To enable customers to access all of Broadcom's SaaS products, Broadcom as a corporation has adopted Okta for single sign-on (SSO). As part of this initiative, API Management SaaS has transitioned its single sign-on solution to Broadcom's **Business to Consumer** (B2C) Okta. This brings parity between API Portal and API Hub login capabilities.
  For information about Okta single sign-on in API Hub, see Access API Hub.
  For information about transitioning to Broadcom Okta Single Sign-On, see the FAQs.
  For more information about Okta, see the Okta site.
  What is Okta?
  Okta is an enterprise-grade identity management service which enables single sign-on and user management to various platforms/applications within the enterprise.

Changed

- **Enhanced View and Search for Portal User List:**
  Portal and Organization administrators can now view ALL users (i.e., Global publishers and organization users) in a single consolidated list when searching for or managing Portal users.
  List filters have also been expanded to include Organization to simplify user management across multiple organizations. The Status filter now includes 'Pending Approval' status to assist Organization administrators to locate such user accounts when the User Registration Request Workflow is enabled.
- **API Hub Updates**
  Multiple updates have been made to the react-admin library to address potential vulnerabilities. View and download the latest API Hub source code from the Layer7 GitHub repository.

Fixed

Issue	Resolution
DE450739	Addressed reports of outdated SSL cipher suites with recommendation of TLS v1.2 protocol.
DE486010	Fixed an issue that prevented a non-Admin role from viewing API specifications and details.
DE488907	Fixed an issue that caused an inconsistent callback length (i.e., limit of 255 instead of 2048 characters).
DE488908	Fixed an issue that that prevented OTK from defaulting an API key to Out-of-Band (OOB) authorization when its API key equivalent in Portal is cleared of any OAuth scope parameters after synchronization.

5.0.0 - 2020-10-20 (S), 2020-10-27 (P)

IMPORTANT

About Staging and Production Environments

Each API Management SaaS upgrade is available to test in a staging environment before it is pushed to production. Refer to the dates in this change log for:

- The date that the upgrade is available in the staging environment (S), and
- The date that the upgrade is pushed to production (P).

IMPORTANT
Note for customers using API Gateway 9.2 or 10 CR1 and higher

For customers using the API Portal with the Layer7 API Gateway version 9.2 (any CR) or version 10 CR1 and higher, review the following information:

- **API Gateway 9.2 (any CR):** There is limited compatibility support when using API Portal 5.0 with API Gateway version 9.2. Do NOT perform update of the integration bundle as part of your 5.0 upgrade. Consequently, a number of new API Portal features will not be available. We recommend upgrading your API Gateway to a later version. See Compatibility Matrix for more information.
- **API Gateway 10 CR1 and higher:** You MUST update the integration bundle and replace the PortalUpgradeAssertion file as part of your 5.0 upgrade. See Update the Integration Software on the API Proxy for more information.

Added

- **Manage Policies with Gateway Bundles**
  Policy authors can create policy bundles in the Gateway Policy Plugin and incorporate them into the CI/CD pipeline for easier upgrades and migrations. As Portal Admin, you can import Gateway bundles into the API Portal and deploy them to the proxies managed in API Portal. This enables Portal as the single source of management console for the lifecycle of API Management to include APIs, Applications, Policies (bundles), and other entities.
  For bundles that are of encapsulated assertion type, these bundles are made available as policy templates for reuse during API publishing. Newer build revisions of these encapsulated assertion type bundles can be uploaded to the Portal for deployment to proxies, gaining automated reuse by the existing APIs leveraging the bundle's policy template.
  For more information on creating Gateway bundles, see the Gateway Policy Plugin documentation.
  For more information on managing policy templates and Gateway bundles in API Portal, see Manage Policy Templates and Manage Policy with Gateway Bundles.
- **New API Portal Dashboard and Menu Bar**
  The new and enhanced API Portal Dashboard provides a quick overview of the current state of your API Portal. Upon logging in to API Portal, you see a personalized view of menu options, entities and analytics depending on your role or customization.
  For more information, see API Portal Dashboard.
- **API Hub**
  - **API Hub: Manage Applications**
    Portal Admin, API Owner, Org Admin, and Org Publisher can now manage applications using API Hub.
    Applications that you add using API Hub are available in API Portal. Note that global publishers (Portal Admin and API Owners) can also continue to create and manage applications in API Portal.
    For more information, see Manage Applications using API Hub.
  - **API Hub: Remote Hosting**
    You can now host your customized API Hub using your own hosting solution. Ensure that you have properly defined your hosting environment in your configuration. Work with your Portal Admin to define the configurations for your customized API Hub in the `config.js` file.
    Following that, you can register the hosting domain of your customized API Hub with the Layer7 API Developer Portal.
    For more information on registering the hosting domain, see Customize and Extend the standard API Hub.
    For more information on defining this file for a customized API Hub, see the the API Hub Example App GitHub page.

Changed

- **Global Theme and Page Appearance Updates**
  In conjunction with the new API Portal look and feel, your theme and page appearance experience has been upgraded.
  The following properties have been added to themes:
  - Color: UI Background color, API Hub primary color, API Hub secondary color
  - Typography: API Hub font
  - Font size: Title, Section title, Small title, Label, Small label

  The following properties are no longer supported:
  - Navigation
  - Header and footer
  - Background image

Upon upgrade, your global theme will include a number of new properties from the default theme. Additionally, the default values for some existing properties have been changed. Changes are automatically applied; validate your settings for any impact on your existing global theme and appearance.
For more information, see Customize Page Appearance.

Deprecated

- **Portal.svc Resource**
  Customization for Sign Up and Account Setup pages are made by making calls to Portal.svc endpoints for payload delivery, query, and validation.
  The following changes have been made:
  - /admin/Portal.svc/Registrations endpoint now validates the unique organization when a new user signs up for an account.
  - /admin/Portal.svc/accountSetup endpoint now validates the unique username when a new user completes the account setup.

  The following endpoints no longer need to be explicitly called have been deprecated:
  - /admin/Portal.svc/OrganizationNameUnique
  - /admin/Portal.svc/UserNameUnique
- **(Imminent) API Explorer**
  The API Explorer application will be deprecated in a future release. You will be able to access Portal's native APIs (including PAPI, Portal Metrics API, Login API, and Authorization API) as individual APIs from the menu bar. This will also give you the ability to view and test the APIs using the Swagger UI
- **(Imminent) Adobe CQ5**
  With the launch of API Hub as the one-stop, customizable platform serving your API consumers and developers, support for creating custom pages using Adobe CQ5 will be removed in mid-December 2020.
  For more information on Adobe CQ5 deprecation, see Layer7 API Management SaaS - Adobe CQ5 Deprecation.

Fixed

Issue	Resolution
DE445856	Addressed a vulnerability issue with server information being displayed in 500 error codes.
DE453331	Fixed an issue that caused the Applications list to disappear when adding API to Applications from the API Details page.
DE452335	Fixed an issue that caused the Action button to not appear when adding API to Application in the API Details page.
DE453523	Fixed an issue where you could not access or bookmark the document using the URI from an API document.
DE463137	API Hub: Addressed an issue that caused incorrect text to appear on the home page when switching languages.
DE478547	API Hub: Addressed an issue with /apihub redirecting to port 8443, which resulted in a redirect error.
DE467651	API Hub: Addressed an issue where a successful SAML login was redirecting back to API Portal home page instead of the API Hub home page.

Issue	Resolution
DE465047	Addressed a sync issue where Gateway-published APIs deleted from the Gateway were still showing up in API Portal.

4.5.x

IMPORTANT

About Staging and Production Environments

Each API Management SaaS upgrade is available to test in a staging environment before it is pushed to production. Refer to the dates in this change log for:

- The date that the upgrade is available in the staging environment (S), and
- The date that the upgrade is pushed to production (P).

Check out the change logs for past 4.5.x releases of the API Management SaaS (API Portal):

4.5.5 - 2020-09-15 (S), 2020-09-22 (P)

IMPORTANT

About Staging and Production Environments

Each API Management SaaS upgrade is available to test in a staging environment before it is pushed to production. Refer to the dates in this change log for:

- The date that the upgrade is available in the staging environment (S), and
- The date that the upgrade is pushed to production (P).

IMPORTANT

Note for customers using API Gateway 10 CR1

For customers using the API Portal with the Layer7 API Gateway, a sync issue exists that renders some Portal-published APIs incompatible with API Gateway version 10 CR1. As a temporary workaround, we recommend using other versions of the API Gateway (9.x or 10.0 base) while a fix is being worked on.

Added

- **Support for Multiple API Keys in Applications:**
  You can now add multiple API key/shared secret pairs to an application. In previous releases, if you required separate API keys, the Portal Admin or API Owner had to add applications for each API key that they needed generated. When a Portal Admin or API Owner added an application to API Management SaaS, API Management SaaS auto-generated an API key, a shared secret, and the other settings that you defined for the application. This API key, along with the secret, were the primary mechanisms for controlling access to the APIs that have been added to the application.
  The application deployment and management process has now been simplified. Now, the API key that API Management SaaS auto-generates when a Portal Admin or API Owner adds an application is the default API key for the application. For existing applications, the API key becomes the default key.
  For more information about the business scenarios for when you should consider managing multiple API keys in a single application, see Manage API Keys.
- **Add Multiple API Keys per Application and Analytics Enhancements:**
  You can now filter and group API hits by API key using the new **API Key** dimensional filter in a custom report.
  For more information:

  - About the updated Swagger file, see Portal Metrics API and Metrics Query API.
  - About this dimensional filter, see Custom Report.
- **New Okta Single Sign-on for API Management SaaS:**
  To enable customers to access all of Broadcom's SaaS products, Broadcom as a corporation has adopted Okta for single sign-on (SSO). As part of this initiative, API Management SaaS has transitioned its single sign-on solution to Broadcom's **Business to Consumer** (B2C) Okta. This enables API Management SaaS to provide a unified experience and a consistent way to authenticate across multiple identity providers (IdPs).
  Users of API Management SaaS do not have to remember which authentication scheme to use in their login as API Management SaaS handles the disambiguation behind the scenes based on the email domain in Security Assertion Makup Language (SAML) SSO configuration. Your users will continue to authenticate with your own IdP, using the same SSO credentials they currently use.

    IMPORTANT
    You can configure only the default login approach to be "SAML" and cannot choose the default IdP within the SAML.

  For more information about SAML SSO, see Configure SAML Single Sign-On.
  For information about transitioning to Broadcom Okta Single Sign-On, see the FAQs.
  For more information about Okta, see the Okta site.
  What is Okta?
  Okta is an enterprise-grade identity management service which enables single sign-on and user management to various platforms/applications within the enterprise.

Changed

- **Testing APIs using an API Key:**
  When testing and exploring an API using Swagger UI, on the **Spec** tab, the Portal Admin or API Owner must now select an application, and then select a specific API key.
  In previous releases, Portal Admins and API Owners had to update their Swagger manually after changing the deployment. OpenAPI 3.0 supports multiple hosts, and continues to require the user editing the Swagger to enter the right hosts. When the secret is hashed, Portal Admins and API Owners must enter the secret manually.
- **Portal integration software and analytics for multiple API keys:**
  If you have metrics enabled (you are tracking analytics) and you are managing multiple API keys for your applications, to have API Management SaaS properly reflect and record the analytics for all API keys including the default key, complete the following:
  - (Hybrid customers only) Update the Portal integration software on the API proxy.
    For more information about how update your integration software, see Update the Integration Software on the API Proxy.
  - (SaaS customers) Contact Support to update your Broadcom-managed proxies.

Deprecated

- **Generate new shared secret endpoint:**
  The `GET /api/<TenantID>/GenerateNewSharedSecret` endpoint for generating new shared secret for API keys has been deprecated. API Portal now uses the following endpoints to generate shared secrets:
  - `POST api-management/1.0/applications/{uuid}/api-keys/`
  - `PUT api-management/1.0/applications/{uuid}/api-keys/{apiKey}`

  For more information, see Portal API (PAPI).

Fixed

Issue	Resolution
DE445823	Addressed an XSS vulnerability involving generation of application keys and secrets.
DE445974	Addressed a user enumeration vulnerability involving application operations.
DE474916	Fixed an issue with application sync to proxies.

4.5.4 - 2020-08-04 (S), 2020-08-11 (P)

IMPORTANT

About Staging and Production Environments

Each API Management SaaS upgrade is available to test in a staging environment before it is pushed to production. Refer to the dates in this change log for:

- The date that the upgrade is available in the staging environment (S), and
- The date that the upgrade is pushed to production (P).

Changed

- **Custom domain length limit:**
  A hard limit to the length of a custom domain name is implemented, default cap is now set to 100 characters.

Fixed

Issue	Resolution
DE467058	Addressed an issue where dispatcher failed to start due to a long custom domain name. A hard limit to the length of a custom domain name is implemented, default cap is now set to 100 characters.
DE459991	Addressed an XSS vulnerability.
DE465764	Addressed an issue with duplicate ApiID when copying over a policy of a Gateway-published API. Previously, multiple Gateway-published APIs could have the same ApiID, which caused sync and API count inconsistency.
DE467365	API Hub: Fixed an issue with an unspecified error message shown when generating a new hashed secret for a proxy with an incompatible OTK version.

4.5.3 - 2020-07-07 (S), 2020-07-14 (P)

IMPORTANT

About Staging and Production Environments

Each API Management SaaS upgrade is available to test in a staging environment before it is pushed to production. Refer to the dates in this change log for:

- The date that the upgrade is available in the staging environment (S), and
- The date that the upgrade is pushed to production (P).

Added

- **API Hub Enhancements:**
  - Portal Admins can now publish generic custom documents in API Hub in the Wiki area. These wiki documents are available to your publishers and consumers. You can also manage these documents using the Portal API (PAPI) Documents resource, setting the type attribute to 'custom' and the typeUuid attribute to 'wiki1'. For more information about how to manage wiki documents, see Manage Wiki Documents in API Hub.
  - Users can now perform account management (such as account signup/setup and reset account password) from API Hub. For more information, see API Hub.
  - Users can now sign in to API Hub using a configured authentication scheme. For more information about authentication schemes, see Configure Authentication Schemes.
  - Users can now use the Healthcare Developer Center as an API Hub example that they can customize and extend. This API Hub example demonstrates a customized API Hub with a Healthcare theme. It extends the standard API Hub to include custom pages and additional PAPI and Portal Metrics API calls. For more information about customizing and extending the standard API Hub, see Customize and Extend the Standard API Hub.
- **Cross-organizational analytics for Org Publisher role:**
  To understand the consumption of APIs across organizations, the analytics reports have been expanded to allow Org Publishers to access metrics and visualize the data for all the APIs and applications owned by their organization as well as applications of other organizations using their APIs. As such, Org Publishers can now filter data by organization and proxies.
  For more information, see Monitor.

Changed

- **The Portal API (PAPI) `applications` resource:**
  The `applications` resource has been enhanced to return the application's description. For more information about this resource, see Portal API (PAPI).

Fixed

Issue	Resolution
DE445829	Addressed a vulnerability issue with default Nginx error pages.
DE445971	Addressed a vulnerability issue with session cookies.
DE462630	Fixed an issue where creating an application with over 50 APIs returned a 500 error.
DE463869	API Hub: Fixed a line wrapping issue on Home page markdown editor.
DE463897	API Hub: Fixed an incorrect locale issue on the Home page markdown editor when switching between languages.
DE463104	API Hub: Removed a non-functional underline icon in the markdown editor.
DE459472	UI: Fixed an issue with the UI not refreshing after deletion of an api-documentation document with many children.

4.5.2 - 2020-06-09 (S), 2020-06-16 (P)

IMPORTANT

About Staging and Production Environments

Each API Management SaaS upgrade is available to test in a staging environment before it is pushed to production. Refer to the dates in this change log for:

- The date that the upgrade is available in the staging environment (S), and
- The date that the upgrade is pushed to production (P).

IMPORTANT

Note for customers using API Gateway 10 CR1

For customers using the API Portal with the Layer7 API Gateway, a sync issue exists that renders some Portal-published APIs incompatible with API Gateway version 10 CR1. As a temporary workaround, we recommend using other versions of the API Gateway (9.x or 10.0 base) while a fix is being worked on.

Added

- **API Hub:**
  Introducing API Hub, a react-admin-based implementation for the developer console of Layer7 API Developer Portal (API Management SaaS). The standard API Hub is a reference implementation that includes localization support and is included with API Management SaaS. You can use the standard API Hub, or you can provide a customized API consumer-facing user experience of API Management SaaS by customizing and extending it. You can extend by adding custom pages, by adding branding/theme changes, and by adding content in the languages that API Hub supports.

  NOTE
  You can deploy and host your customized API Hub on your own domain in an upcoming release.

  For more information, see API Hub.
- **The Portal API (PAPI) Documents resource:**
  You can manage API documents and custom content in API Hub using the `Documents` resource for the PAPI.
  For more information about this resource, see Portal API (PAPI).
- **Application deployment to proxies using API Portal:**
  You can now manage the key deployment type for proxies using API Portal. You can now manage application deployments to on demand proxies using API Portal.
  For more information:
  – See Manage API Key Deployments to Proxies.
  – See Manage Proxies.

Changed

- **Add/Edit API wizard:**
  The Add/Edit API wizard has been enhanced for a more unified end-to-end API management. Publishers can now manage these configurations from the Add/Edit API wizard:

- API Details, including API definition and proxy configuration
- Custom Fields (if enabled)
- Policy Templates
- Spec Authentication (REST API only)
- Management Permissions
- Visibility Permissions
- API Tags
- API Publish State (previously part of API Details)

  NOTE

  - Some configurations, such as Proxy Configuration and Spec Authentication, are not applicable to Gateway-published APIs.
  - As a result of these enhancements, your old bookmarks might not work.

For more information, see Create and Set Permissions for APIs.

Fixed

Issue	Resolution
DE460953	Fixed an issue with custom fields not being returned in the PAPI call if it contained no value.
DE450749	Fixed a vulnerability issue with Missing Secure Attribute in Encrypted Session (SSL) Cookie.
DE455119	Previously, updating API key was enabled through PAPI. However, the new API key failed to respond to requests. As API key update through PAPI is not currently supported, the update request will be rejected.
DE458110	Fixed an issue where the API Spec page only displayed the first 20 registered apps.
DE461404	OrgPublisher can now add and edit docs.

Removed/Deprecated

- **Business Reports:**
  As per the announcement made in October 2019, Business Reports are now officially deprecated and are no longer accessible.
- **Adobe CQ5:**
  This is a preliminary announcement that support for Adobe CQ will be deprecated. The support for creating custom pages using Adobe CQ5 will be removed in mid-December 2020. Customers are advised to leverage API Hub's theming and customization capabilities going forward.

4.5.1 - 2020-05-01 (S), 2020-05-20 (P)

IMPORTANT

About Staging and Production Environments

Each API Management SaaS upgrade is available to test in a staging environment before it is pushed to production. Refer to the dates in this change log for:

- The date that the upgrade is available in the staging environment (S), and
- The date that the upgrade is pushed to production (P).

Fixed

DE456459: Fixed an issue where publishers were not able to perform edit on Gateway-published APIs. Editing APIs now perform as expected.

4.5.0 - 2020-05-02 (P)

IMPORTANT

About Staging and Production Environments

Each API Management SaaS upgrade is available to test in a staging environment before it is pushed to production. Refer to the dates in this change log for:

- The date that the upgrade is available in the staging environment (S), and
- The date that the upgrade is pushed to production (P).

IMPORTANT

Note for customers using API Gateway 10 CR1

For customers using the API Portal with the Layer7 API Gateway, a sync issue exists that renders some Portal-published APIs incompatible with API Gateway version 10 CR1. As a temporary workaround, we recommend using other versions of the API Gateway (9.x or 10.0 base) while a fix is being worked on.

Added

- Manage Proxies for Federated API Deployment
  Portal admins can now assign organizations to proxies, limiting deployment of the organization's APIs to only the proxies allocated for that organization. For more information, see Manage Proxies and Manage API Deployments.
- Assign Managing Organization
  You can now assign managing organization to your APIs. While setting up who can manage an API, you have the option to specify a managing organization or at least one API Owner. Selecting a managing organization allows all users within that organization to edit an API. For more information, see Create and Set Permissions for APIs.
- New Organization-bound Org Publisher Role
  Portal admin can now assign a user as an "Org Publisher" to grant them API publishing permissions within their organization. The Org Publisher belongs under the Org User user type category.
  This role is identical to the API Owner role, but permissions are contained in the organization that the Org Publisher is assigned to.
  - For an overview on the updated roles and permissions, see Get Started - Roles and Permissions.
  - For an updated workflow on assigning user roles, see Manage Users.
  - For more specific information on the Org Publisher role, see Get Started - Roles and Permissions.
- Manage Application Deployments to Proxies
  Portal admins can now deploy applications, identified by API keys, to specific proxies. These proxies represent specific environments and define the backend Gateways. In this way, these Gateways can process incoming requests from physical applications.
  You manage application deployments to proxies by making calls to the `API Key Deployments` resource in the PAPI.
  You can now view the status of application deployments on the new **Deployments** tab within an application. This information includes the status of the deployment, messages about the deployment, and the proxies to which the application (the API key) is deployed.
  For more information about how to manage application deployments to proxies, see Manage API Key Deployments to Proxies.
- Hash Client Secret

To improve the security of the application's client secret in addition to encryption, you can now configure Portal to store the shared secret hashed. An administrator can choose to protect an application's client secret (shared secret) by using a selected hashing algorithm. The hashed secret is initially available for copying when generated, then subsequently displayed and stored in its hashed format. The option of generating plaintext client secrets is still supported for testing purposes, but can be disabled.

For more information, see Enable Hashed Client Secret.

NOTE
Applications created with client secret hashing enabled cannot be synced on proxies running versions of OTK 4.3 or earlier.

- Manage API Documents

  Portal admins, API Owners, or Org Publishers (for APIs that are assigned to their organization) can now add markdown content to your API as API documents. Use these documents to provide more business context about your API and to include other information such as use cases, usage guidelines, and performance details that are useful to your API consumers. These documents are in addition to the Swagger API documentation that is available on the **Spec** tab within an API. You can manage API documents on the **Documentation** tab within an API.

  NOTE
  You can also manage your documents by making calls to the `/document-service/` resource in the PAPI.

  For more information about how to get this Swagger, contact Support.

  For more information, see Manage API Documents.

- Tag API

  Administrators can now associate tags to an API. Tagging helps in grouping and managing APIs so that you can discover APIs based on tags. You can create these tags in Portal or import them from Swagger.

  For more information, see Manage APIs and Create and Set Permissions for APIs

Removed/Deprecated

- Removal of Deprecated API Management Resource

  A new set of API operation endpoints (labeled as `api-management/1.0/apis` in Portal API (PAPI)) were introduced to allow new and future functionalities such as SOAP support, API visibility for Orgs, granular permissions, and updated user roles. These resources are accessible through the APIs page by default or manually through PAPI.

  NOTE
  Earlier API resources (labeled as `/2.0/Apis and /Apis`) were deprecated.

  For more information about the Portal API, see Portal API (PAPI).

4.4 and Earlier

Release 4.4

New API Management Endpoints

A new set of API operation endpoints (labeled as `api-management/1.0/apis` in Portal API (PAPI) 4.4) have been introduced that allow functionalities such as SOAP support, API visibility for Orgs, and granular permissions. These endpoints are accessible through the APIs page by default or manually through PAPI.

Note that earlier API endpoints (labeled as `/2.0/Apis` and `/Apis`) have been moved to the Deprecated section in PAPI 4.4 and will be removed in the next release. We recommend moving to the new endpoints to make full use of API Portal's current and future functionalities.

For more information, see Portal API (PAPI).

Assign APIs to Multiple Organizations

APIs can now be directly assigned to multiple organizations without being attached to an Account Plan. The visibility of APIs is no longer managed on the Account plan level. Only quota and a rate limit are defined on the Account Plan level. For more information, see Manage APIs.

> **NOTE**
> Account Plans no longer control the visibility of APIs. If you have been using Account Plans for indirect API assignment, the organization mapping is updated.

Limit Visibility of APIs to Private, Public, and Restricted

Administrators can now manage the visibility and editing of APIs on the organization level. Admins assign which organizations have private, public, or restricted visibility of APIs and which organizations can manage specific APIs. For more information, see Manage APIs.

SOAP API Support

API Portal now supports publishing and management of SOAP-based APIs. A publisher can publish a SOAP API through Portal or Gateway for developers to discover and consume. For more information, see Manage APIs.

Improved Analytics Solution

Portal installation now deploys an internal Apache Druid analytics engine. For more information on the new analytics solution, see Monitor.

Test and Explore APIs using Swagger UI

Publishers can now access an API's OpenAPI specifications directly from the API Details page. Using the Swagger UI, test and explore the API's resources with improved rendering capabilities. For more information, see Test and Explore APIs.

Use Rich Text in Swagger Files

Conforming with OpenAPI Specifications 3.0 (Swagger) standards, API Portal now supports CommonMark markdown, allowing you to include searchable rich text and media files in your API specifications. For more information, see Edit and Delete APIs.

Integrate with Runscope

You can now integrate with Runscope to create API monitoring tests from your API in API Portal. Runscope is now part of the BlazeMeter Continuous Testing platform. API monitoring from API Portal improves the API lifecycle management. It also allows you to have better visibility into monitoring the performance of key uses cases of your APIs and allows you to quickly identify problems with your API.

The API Portal installation now includes the `integration_core` and the `integration_runscope` databases for the integration. The integration uses the databases only after you enable integrations. The `integration_core` database tracks the integrations in API Portal, such as the Runscope integration. The `integration_runscope` database is specific to the Runscope integration.

Release 4.3.2

Assign API Plans

Publishers can now use API Plans to control how APIs can be consumed by developers and applications within an organization. An API Plan comprises rate limit and/or quota information, along with the public or private APIs that these controls apply to.

For more information on enabling API Plans and its prerequisites, see Manage Plans and Working with API Plans.

Enhanced API View and Search

Manage APIs more intuitively using improved display and search capabilities. APIs can now be viewed as cards with essential information such as state, version, applications, and other relevant metadata, while the new search function supports auto suggest, recent searches, as well as filtering and sorting. For more information, see Manage APIs.

Search User Using Name and Email Address

Search users (Publishers or Developer type) using their name and email address from the Users page. For more information, see Manage User.

Mutual SMTP Authentication between Client and Portal

Client and API Portal can mutually authenticate each other using their corresponding trusted certificates. When configuring external mail server, you can upload the client certificate so that both client and API Portal can be mutually authenticated, providing a better and trusted connection.

For more information, see Configure SMTP at the Tenant-level Configure and Use External Mail Server.

Assign IdP users to Multiple Organizations and Roles through PAPI

Using the updated Portal API (PAPI) Swagger file, administrators can now assign IdP users to multiple organizations and roles before the first login. For more information, see Portal API (PAPI).

Support for Gateway Enrollment with Prefixed OTK

API Portal now supports enrolling Gateways with a prefixed OTK.

Release 4.3

Custom Domain Names

The Administrator can change the default URL to a custom domain name. For more information, see Set Up Custom Domain Names.

Release 4.2.10

Audit Logs

With audit logs, the Administrator can now see a history of actions that are performed on certain objects. The Administrator can also view and sort audit logs in the UI or using an API call. Audit logs can be filtered and exported using an API call. For more information, see Audit Logs.

Map Developers to Multiple Organizations

Administrators can now assign developers to multiple organizations and corresponding roles. This applies to developers who were added in Portal as well as developers who log in to Portal using an external authentication scheme. For the externally authenticated developers, the Portal administrator sets the authorization type of the corresponding authentication scheme to "Portal". The authorization type field is added in this release and is available while adding or editing the user authentication scheme.

For more information, see:

- Manage Users
- Configure Authentication Schemes
- Map IdP Users to Multiple Organizations

Create Custom Core Pages

Replace the default core pages with your own custom pages.

You can now create custom pages using the Adobe CQ5 web content manager and JavaScript code snippets provided in a zip file. The code snippets include graphic elements as well as providing the existing functionality found on the default core pages, such as login/logout state detection and password encryption. From the Content Management page, select your new custom page to replace any of the default core pages such as Home or Login. Alternatively, you can use the custom page as a new page and add a link to it from modified core pages. For more information, see Customize Pages.

Custom Header and Navigation

All custom pages created in Adobe CQ include a standard header and footer format with navigation elements. The header and footer cannot be customized. However, you can hide the header and footer, then provide custom navigation links. Use JavaScript code snippets to create a custom header and footer. For more information, see Custom Header and Navigation.

Release 4.2.9

Portal Page Customizations

You can now customize the following page elements:

- Display portal product version on the login page
- Add images to page footers and headers
- Enable/disable the CA copyright in footers
- Add custom fonts including Web Open Font Format and TrueType
- Change the text labels in the **Publish** navigation menu (APIs, Apps, API Catalog, API Explorer)

For details, see Customize Page Appearance.

Display Proxy URL for the API

You can now display the Proxy URL for the API on the Developer console's View Documentation page. For more information, see View All APIs and Applications on the Developer Console.

Display the List of IDP Users

You can now see the exhaustive list of all external IDP users on the **Administration**, **Users** page. You can view the details of these users, but cannot edit them.

Set Limit on Password Recovery Attempts

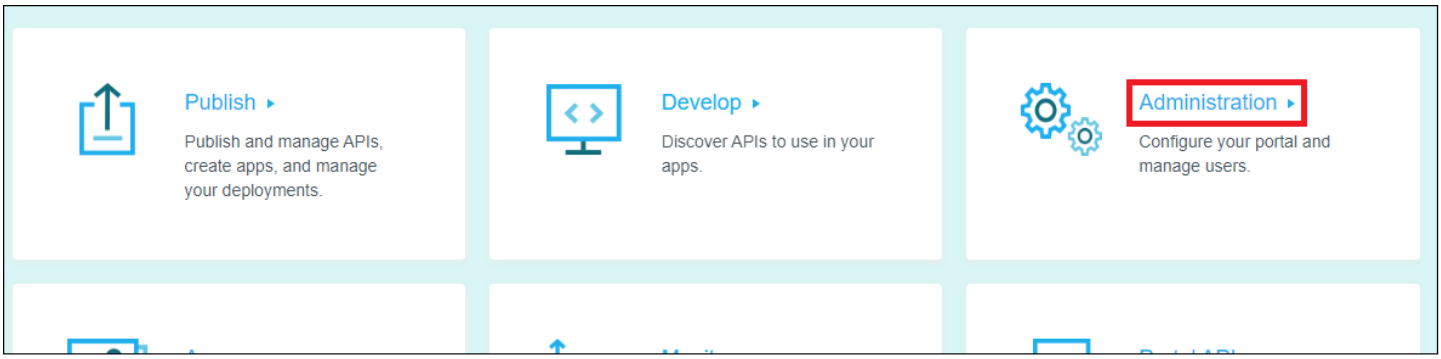
Administrator can now limit the number of auto-generated emails sent to the user, for multiple attempts to recover the password.

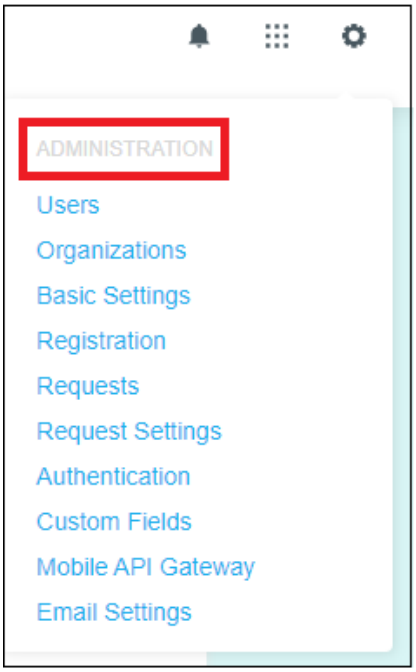
Prevent Multiple Email Registrations

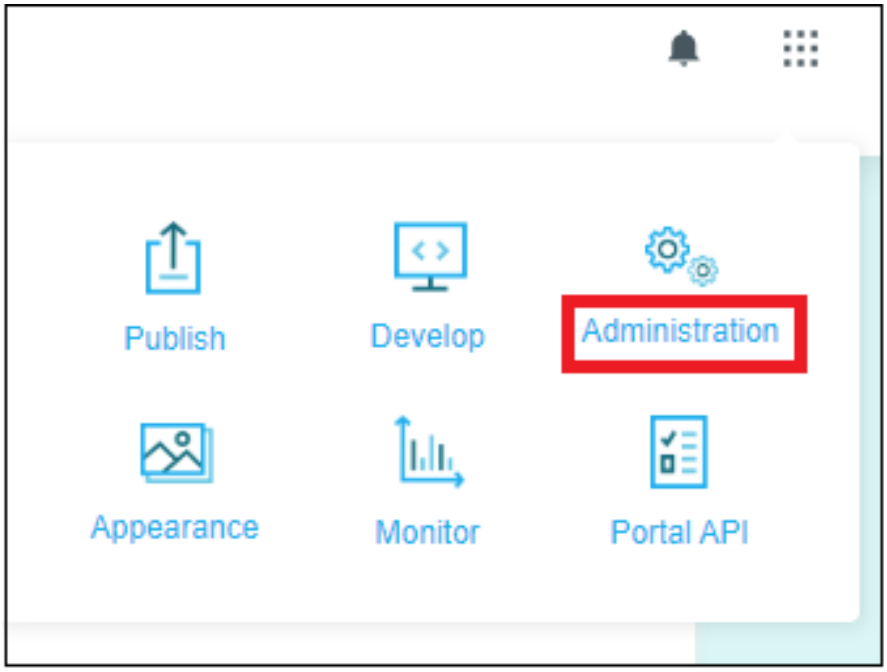
Administrator can now ensure that users first activate their account using the registration email, before sending another request.

Renaming of Settings Menu Name and Settings Icon Name

The **Settings** menu name and the **Settings** icon name has been renamed to **Administration**.


Publish
Publish and manage APIs, create apps, and manage your deployments.
Develop
Discover APIs to use in your apps.
Administration
Configure your portal and manage users.


ADMINISTRATION
Users
Organizations
Basic Settings
Registration
Requests
Request Settings
Authentication
Custom Fields
Mobile API Gateway
Email Settings


Publish
Develop
Administration
Appearance
Monitor
Portal API

Release 4.2.8

Encrypt User Credentials Over Non-SSL Connections

For LDAP and CA APIM (default) authentication schemes, the administrator can improve security by encrypting user credentials over SSL connection. Enhanced security reduces the man-in-the-middle attacks by encrypting the user passwords. For more information, see the following sections:

- Configure Lightweight Directory Access Protocol
- Configure Authentication Schemes

Update to the Organization Drop-Down List

The Organization drop-down list in the API Details tab on the Publish an API page has been improved so that the list displays 10 results at a time, displaying the next 10 results when you reach the end of the list. It also allows you to enter text so you can do a keyword search.

Defect Fixes

This release corrected a number of defects.

Release 4.2.7.2

Defect Fix

This release corrected a SAML SSO login defect.

Release 4.2.7

Filter Views for API Groups, Applications, Organizations, and Account Plans

The usability and viewing of API groups, applications, organizations, and account plans has been improved.

You can now:

- Filter APIs while you are adding or editing API groups and account plans. The APIs are now paginated.
- Filter by API or API group while you are adding or editing an application. The APIs are now paginated.

In addition, organizations are now paginated while you are adding an application.

Set Password Expiry and History in the Password Policy

Administrators can now configure password policy to set the password expiry duration, and also restrict the reuse of the old passwords.

For more information, see the Manage Password Policy section.

Release 4.2.5.3

XSS Vulnerability Fixes

Release 4.2.5.1

SAML Configuration Updates

SAML 2.0 is an XML-based protocol that uses security tokens to pass user authentication and authorization data between an IdP, and a service provider. CA API Developer Portal uses user authentication when integrated with SAML IdP system. Issuer ID is the new configuration parameter for SAML, and the Service Provider ID is a mandatory value.

For more information, see the Configure SAML Single Sign-On section.

API Sync Improvements

Improved API sync mechanism from tenant Gateway perspective. Enhancements include:

- Reduced sync times in both medium- and large-scale deployments, as well as when APIs are introduced and/or modified.
- Addressed API and API fragment duplication issues.
- Addressed memory error in large-scale deployments.
- Addressed database retrieval timeout issue.
- Added the following internal reserved custom field names:
  - 'PortalModifyTS'
  - 'PortalID'

NOTE

To activate these sync improvements, you will need to update the API Portal integration software. This update is required in both SaaS and hybrid deployments. See *Update the Integration Software on the API Proxy* in Integrate On-Premise API Proxies.

Release 4.2.4

Secure Socket Layer (SSL) Support in LDAP Authentication

Administrator can now configure CA API Developer Portal to support LDAP with or without SSL for user authentication.

For more information, see the Configure Lightweight Directory Access Protocol section.

Release 4.2.3

Configure Search Expression in LDAP Authentication Schemes

Administrator can now configure LDAP authentication schemes to include user account that has privileges to search for users, and specify search expressions for locating users in LDAP directory.

For more information, see the Configure Lightweight Directory Access Protocol section.

Active Directory Integration for Logging In to API Management SaaS

Administrator can now enable Active Directory user to log in to API Management SaaS with samAccountName as a login attribute.

For more information, see Configure Microsoft Active Directory section.

Increased Stability and Data and UI Performance

Further improvements are made to increase stability and performance when working with a larger number of organizations and APIs.

Release 4.2.2

Improved Load and Response Time

Improvements are made to reduce load and response time when viewing and working with a larger number of organizations and APIs.

CA Gateway 9.3 and OTK 4.2 Support for Hybrid Deployments

In this release, CA API Portal supports Gateway 9.3 and OTK 4.2 compatibility for Compatibility Matrix.

Release 4.2

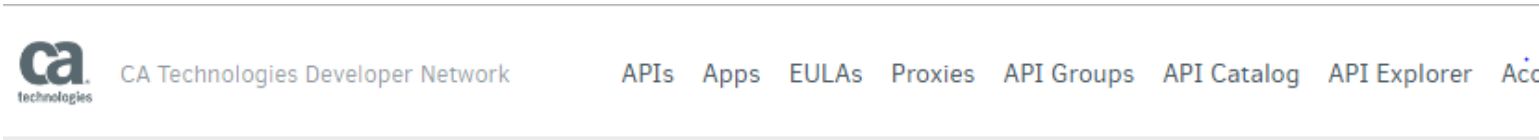
Configure Password Policy

Administrators can now configure a set of rules to create complex passwords. For information about how to modify password policies, see the Manage Password Policy section.

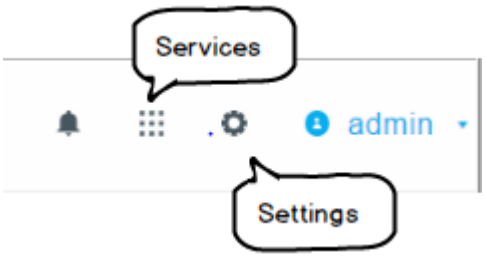
CA API Management Console

The new CA API Management console provides a consistent experience for all roles.

- **A new dashboard**
  The dashboard overlays the existing Portal functionality. No existing Portal functionality has been removed; you'll just find services in a different place.
- **New global header**A global header replaces the existing main menu for improved navigation.
  For example, here is the global header navigation for the **Publish** service.



- Use the **Services** and **Settings** icons for faster navigation between pages.



- **New Developer Service**
  We've added an intuitive workflow for App developers to discover and consume APIs.
  On the dashboard, select **Services**, **Develop** and then etiher of the following:
  Discover APIs
  – Search APIs (improved performance)
  – View all APIs
  – View Swagger file or download it to use in your development tools
  Consume APIs

  - Add API to app
  - Configure app (get API key/shared secret)
  - Manage app
- **New Appearance Service**
  Admins can customize global themes at the tenant level.
  On the dashboard, choose **Services**, **Appearance, Manage Global Theme,** and then change...
  - Header logo and console name
  - Fonts and sizes
  - Colors

Q. How do I get the new console?
A. Simply upgrade to 4.2. No configuration is required.

Q. How do roles map to the new console functionality?
A. Roles and Permissions.

Q. Any general limitations of the new console?
A. Just a few, but stay tuned for updates!

- – When searching APIs, you can search only by **Name** and **Description**.
  – API documentation is only in Swagger (we know you'd like to add your own custom documentation).
  – New apps that are pending approval are not displayed in the console.

Publish API without Deploying it in an Environment

An API can be created, retrieved, updated, or deleted without explicitly deploying. The Details page displays which published APIs were not deployed.

Automated Event Driven Targeted API Deployment

- **New API Deployment Management**
  New API deployment allows any user with create, read, update, and delete permission to author an API until it is ready for deployment to one or more environments using the new deployment types from a single Portal.
- **Deployment Type Setting on Proxy**
  Administrators and API owners now have the ability to publish an API and deploy it to a runtime environment for the following deployment type options:
  - Automatic
    API are automatically deployed once they are published or updated.
  - On-demand
    API deployments are deployed on-demand by calling the deployment APIs. These APIs can be accessed from the Portal APIs link in the navigation menu.
  - Scripted
    API deployments can be integrated into your existing CICD workflow using the deployment APIs and invoking them from your deployment script.

  You select the deployment types when adding or editing a proxy from the following pages:
  - Add Proxy
  - Edit Proxy

  API publishers can perform the following tasks:
  - Enable event-driven deployment of an API to a specific proxy
  - Promote an API across functional environments (dev, test,and prod)
  - Deploy an API to some specific environments (geographies) and not all
- **New API Details Page**

The Details page displays the following information:

- – Name of the API
  – List of proxies where the API is deployed
  – List of proxies where the API is not deployed
  – Sync state for each deployment
  – Deployment type for each API
  – Time and date of the deployment
- **New Deployment API**
  On-demand and scripted deployment use the Deployment API to trigger event-driven API deployments.

  NOTE

  Info: For more information:

  - About the new API deployment, see Manage API Deployments.
  - About the API Details page, see View How and Where an API is Deployed in the Details Page.
  - About the Deployment API, see Deployment API.

Known Issues: API Management SaaS

This article lists the known issues in the Layer7 API Management SaaS as categorized by functionality.

System, Set Up and Configuration

Issue	Description
DE480720	API Hub: When UI developers are customizing and developing API Hub locally and want to use the mock services for local development, the mock service is not working in applications.
DE467987	API Hub: Custom domain registration with the standard API Hub is not currently supported.
DE437796	There is a known issue in Portal system logs where all "INFO" and "WARNING" messages are incorrectly tagged as "ERROR".
DE439214	A Druid bug is causing ingestion failure when the entire stack is restarted, or when there is no data flowing into the ingestion pipeline for more than several hours. The errors can be viewed in the coordinator logs.
DE429876	When a tenant provision and a tenant deletion requests are made within a small time frame, not all records are deleted. The second delete log reports two records deleted from tenant_provisioning database. The expected behavior is for log to report zero deleted records. To resolve the issue, make another DELETE call. The second call will clean up the undeleted orphan records.
DE268541	If the time on the API Portal and tenant Gateway is not synchronized, API Portal administrators might be unable to enroll a tenant Gateway.
DE393129	When a Portal Admin or an API Owner selects the Requests icon and then selects an existing application request, a "404 Page Not Found" error is shown.

API Management

Issue	Description
DE492449	APIs using a policy template created from a Gateway bundle have reportedly failed to deploy to a proxy configured for Automatic API deployment. The workaround is to ensure that the Gateway bundle is deployed to the proxy first before publishing the API using a policy template.
DE491093	A discrepancy in the Gateway Published API count may occur when a Gateway-published API is enabled in Portal but then deleted from the Gateway. To work around this discrepancy, a user must disable and delete the API in Portal - the next synchronization should permanently delete it from the Portal database and API count.

Issue	Description
DE480867	API consumption failures are known to occur for an API consumed by an application with multiple API keys, or if the API is assigned to more than one API Plans. This issue only occurs when using an OAuth 2.0 policy template.
DE480865	API Plans do not sync on a proxy that has been created or enrolled with on-demand API key deployments due to the application sync returning an empty response. As a workaround, manually set the **portal.api.plans** cluster-wide-property to TRUE in the Policy Manager.
DE480730	Download of Swagger files from API Explorer is currently unavailable. As a workaround, view and download the latest sample files from Portal APIs.
DE463842	API Hub: Application credentials aren't pre-filled for Authorization under API Spec.
DE438309	When opening and editing a Gateway-published SOAP API in Portal, the system will throw an exception. This issue occurs when the **Custom Resolution Path** field is empty when publishing a SOAP API through the Policy Manager. As a workaround, ensure that you resolve the **Custom Resolution Path** field when publishing a SOAP API via the Policy Manager.
DE452534	If a legacy automatic proxy is enrolled to the tenant using OTK 4.3.1 or earlier, hashed secret cannot be generated or regenerated even if the application was never deployed to the proxy. The following message appears: "API key was deployed to proxies that have OTK version which does not support hashing secret. Please either use plain text secret or upgrade OTK version to 4.4 or higher of those proxies." Choose one of the following workarounds to enable hashing: • Upgrade OTK to 4.4 or later on the proxies that will manage Applications with hashed secrets. • Use PAPI to set the API key deployment type to be On Demand for proxies whose OTK has not been upgraded.
DE356210	You cannot update or delete API deployments for APIs deployed to Automatic proxies after you change the proxy's API deployment type to On Demand or Scripted. To work around this issue, after you change the proxy's API deployment type from Automatic to On Demand or Scripted, modify the APIs associated with this proxy to force a re-deployment. This includes a change to the following API fields: name, status, application URL, proxy URL, custom fields, or policy templates.
DE353662	If a newly provisioned hybrid tenant is not yet enrolled, Policy Templates list in the Add/Edit API pages will not display correct policy templates. To rectify this issue, enroll the tenant.
DE268849	If a Gateway-published API has a double-byte character in the name, each time the API is synchronized the name is extended until all APIs fail to synchronize.
DE267506	APIs with names longer than 245 characters cannot be synchronized with the tenant Gateway.

Issue	Description
DE391472	When the "Quota by Day" and the "Quota by Month" policy templates are combined, API requests are counted twice. This behavior occurs because each policy template has the "Throughput Quota" assertion evaluated. As a result, API consumption fails due to exceeded quota. To avoid the issue, we recommend that you use only one of the policy templates.
DE391082	If an API gets deleted while it is used in "api-group" and "application", the last application sync shows as failed on the API Proxy page. The sync failure shows due to the inability of the application without an API to sync to the proxy. All other applications continue to work and sync as expected.

Analytics

Issue	Description
DE436220	The Daily Quota Consumption chart's X-axis and tooltip show the data for month and date but does not show the time. If you select the current day data hits, the chart shows "NO DATA".
DE480649	The default API key name is not displayed in custom reports. This issue is normally observed after an API's Account Plan or API Plan quota has been exceeded.
DE479701	When an Org User switches to a different Organization, traffic and custom reports are not showing the newly selected organization data or filters. As a workaround, log out and log back into the API Portal.

UI, Design, and Themes

Issue	Description
DE479446	When new mandatory Custom Fields are added after an application is created, the validation message in Edit Application may not be entirely clear. As a workaround, manually check if there are any required Custom Fields.
DE435918	When you authorize an API and a validation error is shown, you should be able to hide the error message. Clicking the Hide button next to the Error message does not hide the message.
DE425959	The issue occurs in Manage Application. The user is unable to filter by Application State in Application List page. For example, when the admin edits an application name and navigates to Application List page filtered by state, Pending Application is not shown.
DE426270	The issue occurs in Manage Application. The user is unable to filter Rejected Applications. When an administrator rejects an application, the filter with state set to Rejected shows an empty list.

Issue	Description
DE426467	The Application incremental synchronization is not working as expected. The error code 103 from invoking "Portal API Key Sync" prevents the application sync job from updating the increment start time. The portal.application.increment.start is not updated to the last synced time.
DE392603	When an Administrator opens the Users page from **Administration** on the Dashboard, the user list is not sorted by name alphanumerically in the **Name** column. To search for a particular user if there is a long list of users, the Administrator can use the browser's CTRL-F functionality.
DE391920	The issue occurs after an Administrator creates an Application (enabled) and creates a new Application custom field, and then goes to **Develop**, **View All Applications**, and selects the newly created Application. On the **APIs** tab, when you hover over an API and select **Remove**, an internal server error is displayed, the Application is not saved, and a message to provide more information does not display.

Roles and Authentication

Issue	Description
DE481425	API Hub: Password policy with enhanced security options does not work as expected when using Layer7's default authentication scheme provider.
DE479762, DE480908	API Hub: Okta and legacy SAML logins are not currently supported.
DE511795	A "Request method 'POST' not supported" error is presented by the Portal after a password change is attempted by the user. This password error occurs after the user uploads a modified SampleTemplates folder to customize the Portal. For a workaround, see the Knowledge Base article here.

Resolved Issues: API Management SaaS

This article lists the resolved issues in the Layer7 API Management SaaS in chronological order.

Resolved Issues in Release 5.1.2

Issue	Resolution
DE529051	Resolved an issue that caused permission denied error in pssg pod while installing Portal.
DE529634	Resolved an issue that caused an error when accessing information from the portal using PAPI calls simultaneously from multiple tenants.
DE533754	Resolved a special character issue that corrupted Portal pages.
DE534853	Resolved an issue with Gateway performance that involved a Gateway Database call to validate the counters.
DE535688	Fixed an issue with the Content-Security-Policy that raised security concerns when requests were sent to the portalhost.
DE535732	Fixed an issue with deleting organizations from the Portal when one or more users belong to the organization being deleted.
DE535920	Fixed an issue that caused a Gateway response that identifies the Gateway in the Server header.
DE535946	Resolved an issue where input fields weren't validated against malicious content (e.g., JavaScript injections).
DE537070	Fixed an issue where workflows that require approvals prevented the generation and consumption of hashed secrets.
DE537340	Resolved a Portal endpoint issue that caused the 'Referrer-policy : SAMEORIGIN' header to not appear in the response.
DE537426	Resolved a vulnerability where Portal cookies were found to not have a set SameSite attribute.
DE538848	Resolved Portal upgrade issue with login, deletion and recreation of user accounts.
DE539612	Resolved multiple Portal endpoint vulnerabilities, including (but not exclusive to) missing security headers, host header injection, lack of input length control, and potential SQL injection. The Global XSS Filter was introduced to protect the Portal against cross-site scripting attacks.
DE540461	Fixed an issue that prevented a client from retrieving an access token if the 'Add Application Request' workflow is enabled and hashed secret is enabled for application shared secret security.
DE540556	Resolved an issue that caused an unwanted Gateway policy fragment (Route via HTTP) to appear when a SOAP API is exposed via the Portal.
DE540595	Resolved an issue that caused the Portal to route end users from an email link to the main tenant page instead of the intended API Hub customized page upon user activation.
DE540596	Addressed a request to allow customization of the sender's no-reply notification email addresses instead of using the default CA or Broadcom address.
DE542530	Resolved an issue concerning the inability to enforce a rate limit for Portal applications and the APIs contained within.

Resolved Issues in Release 5.1.1

Issue	Resolution
DE503030	Resolved an issue that caused a "Unable to parse attribute time" error when upgrading LAC 5.1 to 5.4.1.
DE511078	Resolved a Portal synchronization issue that showed an incorrect application count in the dashboard.
DE524894	Resolved a synchronization issue where a Gateway-published API was added to an API group and was deleted from the proxy without being removed from the group.
DE525323	Resolved a Portal restart issue in which the provisioner containers start up process timed out with a 'Waiting for Changelog lock" error.
DE525359	Resolved an issue to save and retrieve Kubernetes secrets in external vault during disaster recovery.
DE525407	Resolved an NGNIX issue in which running an OpenSSL query to the Portal returned only a server certificate and not the full certificate chain for an Azure Gateway.
DE527263	Resolved an database issue that prevented the Portal from starting after upgrading to version 5.0 CR1 and MySQL 8.
DE528190	Resolved an outdated web server issue that prevented LAC from integrating with LDAP.
DE528273	Resolved a Druid issue that prevented a Portal upgrade with an external MySQL database from being completed successfully.
DE528585	Resolved an issue that caused approval emails to be routed to the incorrect person or email address.
DE528586	Resolved an issue that prevented the Portal from authenticating the user attempting to send a PUT request when registering the hosting domain of a customized API Hub with the Portal.
DE529513	Resolved a Portal migration issue that produced a 'Gateway Published MM Alerts AMQ is missing on the target portal' error.
DE530115	Resolved a database table issue that prevented a Portal user from updating a Gateway-published API after it synchronizes with the Portal.
DE530429	Resolved an API Hub issue that saw the Portal fetch a non-existent resource due to an incorrectly entered path that led to an ingress validation failure.
DE531297	Resolved a Portal migration issue (migrating from Portal version 3.5 to 5.x) that prevented Callback URL values from being migrated.
DE531910	Resolved an issue to pass authorization within SaaS Portal using identical login credentials.
DE531951	Resolved a problem in which the user is unable to modify visibility or management permissions in the Portal due to MariaDB being incompatible with the Portal.
DE531971	Resolved PAPI call fail issue to update Organization due to unavailability of Organization Type.
DE532004	Resolved an issue with Openshift cluster upgrade that prevents portal from starting due to application sync fail.

Issue	Resolution
DE532169	Resolved a Portal migration issue (migrating from Portal 3.5 to 4.5) that prevented the migration of rejected applications, causing the migration job to be unsuccessful.
DE532788	Resolved an issue that resulted in elimination of log4j jar file instances from Zoeekeper and Kafka libraries.

Resolved Issues in Release 5.1

Issue	Resolution
DE505495	Fixed an issue that prevented a user from clicking the last line of a swagger sample file.
DE505576	Fixed a performance issue that caused Kafka producer to skip some of the messages within the configured timeout (request.timeout.ms) when multiple gateways were sending requests to ingestion-server.
DE507512	Fixed a database issue that prevented a user from deleting an organization (500 Internal Server Error).
DE508099	Fixed an uploader issue that caused swagger assets with non-English (double-byte) characters to appear incorrectly.
DE511455	Fixed a permissions issue that caused custom core pages to load and render incorrectly.
DE517146	Fixed an issue that caused applications to be out of synchronization when the API groups they subscribe to have been deprecated.
DE517396	Fixed an issue that caused the Portal database password to appear in the ingress pod startup logs.
DE518601	Fixed a database issue that prevented the API drop-down list from displaying APIs in API Explorer when there are many tenants configured in the API Portal.
DE519696	Fixed a security issue by displaying a nondescript '404' error and removing any unnecessary IP address details in the returned error message when a user attempts to access the Portal with an unknown host.
DE521042	Fixed an API Hub issue that prevented a user from viewing the bottom line(s) of an API document hidden behind the Broadcom Copyright statement footer.
DE521069	Fixed an API Hub customization issue that caused theme colors to be inconsistently applied.
DE522251	Fixed a sychronization issue that prevented users from viewing or selecting API plans that contained APIs that changed from 'unpublished' to 'enabled' when adding APIs to an application.
DE524608	Fixed an issue that caused multi-byte characters to incorrectly display on the API spec page.

Resolved Issues in Release 5.0.3

Issue	Resolution
DE492129	Fixed an issue that prevented users from entering accented characters (UTF8) in the Portal user interface, such as the entry of API tags.
DE507512	Fixed an issue that prevented users from deleting organizations from the Portal via PAPI.
DE504854	Resolved a ReactJS UI issue that caused some custom fonts to not appear after selecting them when managing global themes.
DE508099	Fixed an issue that incorrectly rendered swagger files that contain non-English (non-ASCII) characters.
DE511445	Fixed an issue that caused custom core pages to not load correctly on the Portal.

Resolved Issues in Release 5.0.2

Issue	Resolution
DE452352	Fixed an issue that enabled an Org admin to perform CRUD operations on document-service endpoints even when their organization does not have managing permissions for those endpoints.
DE453218	Fixed a CORS vulnerability that could potentially allow the unauthorized disclosure of data pertaining to authenticated users.
DE466904	Fixed an issue that caused the API Hub to not provide a warning to developers when specifying duplicate organization names during signup.
DE479561	Fixed an issue that prevented the API Hub from supporting password resets via email.
DE480649	Fixed an issue in the Analytics Custom Reports that was not displaying the default API key name. This issue was normally observed after an API's Account Plan or API Plan quota had been exceeded.
DE481425	Fixed an issue in API Hub that prevented a password policy with enhanced security options from operating correctly when used with Layer7's default authentication scheme provider.
DE488262	Fixed an issue that prevented the /v2/users PAPI endpoint from returning users based on the ORG access token.
DE491093	Fixed an issue that caused Gateway-published APIs to be incorrectly included in the API deployment count.
DE492449	Fixed a synchronization issue that incorrectly displayed a deployment error after a Gateway bundle is successfully deployed to a Gateway.
DE495779	Fixed a migration issue that prevented users from editing API plans (i.e., adding a new API to an API plan) with a message incorrectly informing them to remove a linked API from an application.
DE498160	Fixed a Portal sync application issue that occurs after a classic Portal (i.e., version 3.5 or older) migration.

Resolved Issues in Release 5.0.1

Issue	Resolution
DE450739	Addressed reports of outdated SSL cipher suites with recommendation of TLS v1.2 protocol.
DE486010	Fixed an issue that prevented a non-Admin role from viewing API specifications and details.
DE488907	Fixed an issue that caused an inconsistent callback length (i.e., limit of 255 instead of 2048 characters).
DE488908	Fixed an issue that that prevented OTK from defaulting an API key to Out-of-Band (OOB) authorization when its API key equivalent in Portal is cleared of any OAuth scope parameters after synchronization.

Resolved Issues in Release 5.0.0

Issue	Resolution
DE445856	Addressed a vulnerability issue with server information being displayed in 500 error codes.
DE453331	Fixed an issue that caused the Applications list to disappear when adding API to Applications from the API Details page.
DE452335	Fixed an issue that caused the Action button to not appear when adding API to Application in the API Details page.
DE453523	Fixed an issue where you could not access or bookmark the document using the URI from an API document.
DE463137	API Hub: Addressed an issue that caused incorrect text to appear on the home page when switching languages.
DE478547	API Hub: Addressed an issue with /apihub redirecting to port 8443, which resulted in a redirect error.
DE467651	API Hub: Addressed an issue where a successful SAML login was redirecting back to API Portal home page instead of the API Hub home page.
DE465047	Addressed a sync issue where Gateway-published APIs deleted from the Gateway were still showing up in API Portal.

Resolved Issues in Release 4.5.5

Issue	Resolution
DE445823	Addressed an XSS vulnerability involving generation of application keys and secrets.
DE445974	Addressed a user enumeration vulnerability involving application operations.
DE474916	Fixed an issue with application sync to proxies.

Resolved Issues in Release 4.5.4

Issue	Resolution
DE467058	Addressed an issue where dispatcher failed to start due to a long custom domain name. A hard limit to the length of a custom domain name is implemented, default cap is now set to 100 characters.
DE459991	Addressed an XSS vulnerability.
DE465764	Addressed an issue with duplicate ApiID when copying over a policy of a Gateway-published API. Previously, multiple Gateway-published APIs could have the same ApiID, which caused sync and API count inconsistency.
DE467365	API Hub: Fixed an issue with an unspecified error message shown when generating a new hashed secret for a proxy with an incompatible OTK version.

Resolved Issues in Release 4.5.3

Issue	Resolution
DE445829	Addressed a vulnerability issue with default Nginx error pages.
DE445971	Addressed a vulnerability issue with session cookies.
DE462630	Fixed an issue where creating an application with over 50 APIs returned a 500 error.
DE463869	API Hub: Fixed a line wrapping issue on Home page markdown editor.
DE463897	API Hub: Fixed an incorrect locale issue on the Home page markdown editor when switching between languages.
DE463104	API Hub: Removed a non-functional underline icon in the markdown editor.
DE459472	UI: Fixed an issue with the UI not refreshing after deletion of an api-documentation document with many children.

Resolved Issues in Release 4.5.2

Issue	Resolution
DE460953	Fixed an issue with custom fields not being returned in the PAPI call if it contained no value.
DE450749	Fixed a vulnerability issue with Missing Secure Attribute in Encrypted Session (SSL) Cookie.
DE455119	Previously, updating API key was enabled through PAPI. However, the new API key failed to respond to requests. As API key update through PAPI is not currently supported, the update request will be rejected.
DE458110	Fixed an issue where the API Spec page only displayed the first 20 registered apps.
DE461404	OrgPublisher can now add and edit docs.

Resolved Issues in Release 4.5.1

Issue	Resolution
-	N/A

Resolved Issues in Release 4.5

Issue	Resolution
DE413277, DE436878	Fixed an issue with API Proxy URL where an incorrect port number 8443 was getting added to the URL.
DE442007	Fixed an issue with reports wherein no data was displayed for a certain time period.
DE444428	Addressed a vulnerability issue.
DE445353	Fixed IdP certificate cache issue in SAML login flow.
DE439014	Fixed an issue with API filtering. Previously, when filtering an API by organization that had no API access data, instead of no data, the chart showed results for No Auth Public APIs
DE420227	Fixed an issue with API Search. Search results now return all APIs which match the search criteria entered.
DE433103	Fixed a health check issue wherein portal-data was shown as "unhealthy", even when there was no issue.
DE439106, DE427414	Addressed a vulnerability issue with the 'Echo' service that gets deployed on the Gateway through the Portal.
DE440640	Increased the character limit for Organization name to 255 characters.
DE445818	Addressed a CORS vulnerability issue.
DE445819	Addressed a vulnerability issue.

Resolved Issues in Release 4.4

Issue	Resolution
DE432608	Corrected an issue where the icons on the top navigation bar were not clearly visible when the background color was changed.
DE432606	Corrected an issue where the hyperlinks in API documentation section of the Developers page did not work properly.
DE432604	Corrected an issue where the color of some Portal UI elements could not be changed.
DE424117	Corrected an issue with unreadable texts due to background color.
DE423471	Corrected an issue with header color.
DE423467	Corrected an issue with header text color in Appearance.
DE415969	Corrected an issue where the HTML tags in swagger document were not shown properly in API Explorer, API Catalog.
DE427865	Corrected an issue with a Policy templates string that was limited to 255 characters. When entering more than 255 characters, the API publishing process failed.

Issue	Resolution
DE432972	Addressed a limitation with Swagger without security definition. Pre-defined security definitions in Swagger files are now supported and rendered in the Swagger UI.
DE431491	Corrected an issue where Portal Sync API Scheduled Task updated all Gateway-published APIs and it resulted in increase of memory, CPU utilization, and GC in Gateway.
DE431434	Corrected an issue where Portal user ID login was not case sensitive.
DE424912	Corrected an issue where scope data was cleared and changed back to default when values in applications were edited.
DE421610	Corrected an issue where Japanese text strings in the response get garbled if you send a request to API Gateway via API Explorer.
DE417079	Corrected an issue where only description was visible for some Swagger files in API Explorer. Swagger files are now rendered correctly in the Swagger UI.
DE376724	Corrected an issue with long URLs as external authorization and token endpoints. Authorization and token endpoints can now accommodate strings up to 900 characters.
DE432188	Corrected an issue where the user search by organization was not working in Portal API (PAPI). The calls were returned with all users and no filter was applied. Resolved so that only the users belonging to the organization are listed now.
DE429305	Corrected an issue where the real-time Analytics dashboard showed only HTTP 500 Error in the pie chart even if HTTP 40x errors existed.

Compatibility Matrix

Review the following topics:

Supported Browsers

Developers and API publishers need one of the following browsers to use the API Portal:

- Mozilla Firefox 70 or later
- Google Chrome 76 or later

Support for Layer7 API Management Products

This table compares the compatibility of the SaaS deployment of Layer7 API Developer Portal with Layer7 API Management products.

APIM Product	Supported Versions	Notes	
Layer7 API Gateway	• 10.1 • 10.0 • 9.4	API Portal requires Layer7 API Gateway version 9.4 or higher. To manage API Gateway, API Gateway administrators need that same version of the Layer7 Policy Manager application.	Layer7 API Gateway documentation
Layer7 API Management OAuth Toolkit (OTK)	• 4.5.x	For hybrid deployments, ensure that the Layer7 API Gateway has OAuth Toolkit (OTK) 4.2 or higher installed on it. **Note:** API Portal integration with a Cassandra database requires OTK 4.3 or higher.	Layer7 API Management OAuth Toolkit documentation

Supported Identity Providers

Administrators can now configure, and authenticate Portal users using the following Identity Providers:

- Lightweight Directory Access Protocol (LDAP)
- SAML Single Sign On (SAML SSO)
- CA Single Sign-On

The configured authentication types are available concurrently from the Layer7 API Portal. For information about how to configure the Identity Providers, see Configure Authentication Schemes.

Supported Integration Bundle Software

The following version of the Portal Integration bundle should be installed to leverage and optimize the latest automatic deployments for APIs and Application API Keys:

- UTC.20220919.1200.00

TIP
How To Tell If You Have the Latest Integration Bundle Installed

You can determine the version of the installed bundle by accessing the Gateway Policy Manager and looking up the 'portal.bundle.version' cluster property. The property should have a value of 'UTC.20220919.1200.00' or higher.

Set Up and Maintenance

The tasks in this section are performed by a Portal Admin.

The following articles describe how to set up and maintain the API Management SaaS (API Portal):

Set Up API Management SaaS

When you set up a new API Portal, there are some procedures that administrators must perform, depending on how the API Portal will be deployed and configured. There are other procedures that administrators might prefer to do during setup.

To deploy the API Portal as a hybrid solution:

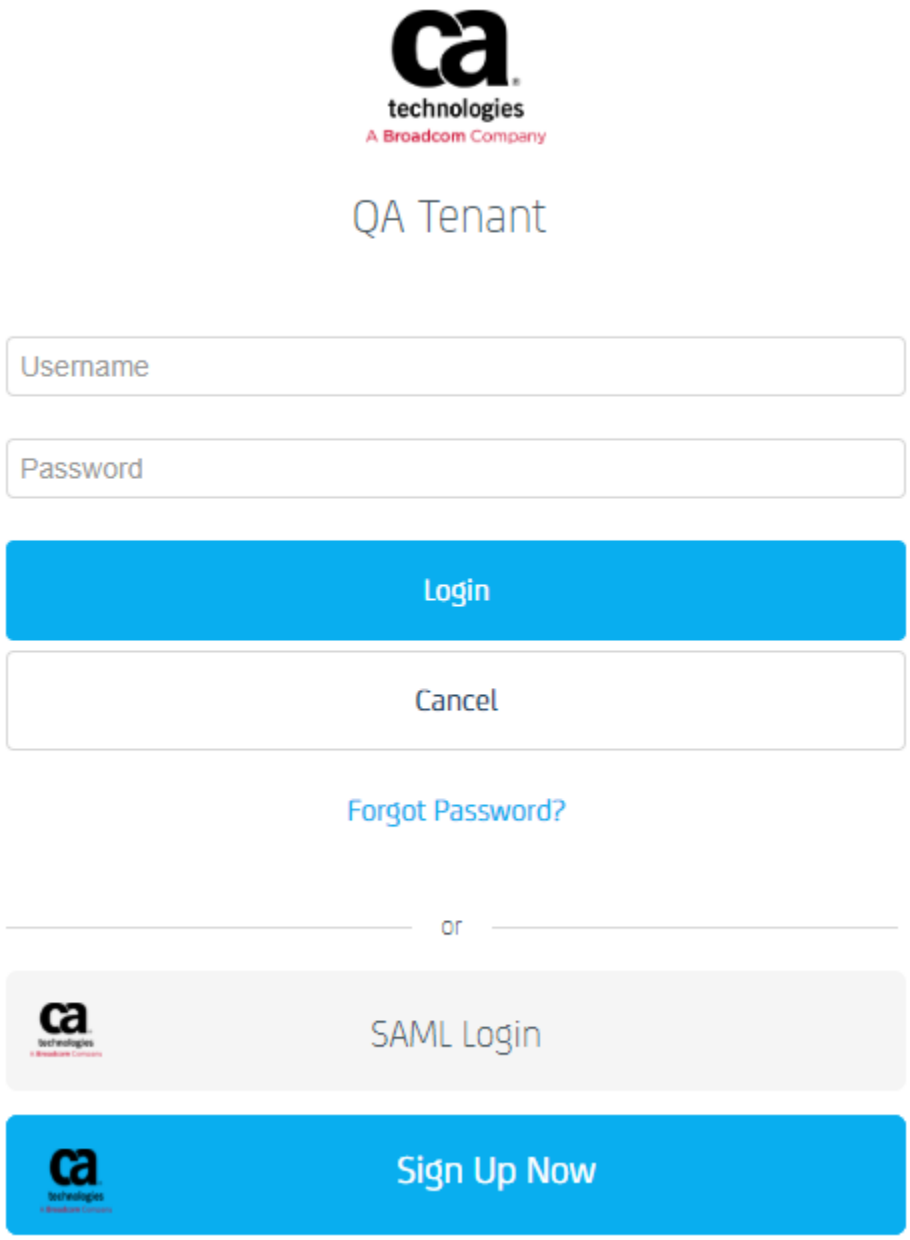
1. An API proxy administrator must install and configure their on-premise API proxy. See "Install and Configure the Gateway" in the online documentation for the API Gateway.
2. An API proxy administrator and an API Portal administrator must integrate the API proxy with their instance of the API Portal. See Integrate On-Premise API Proxies.

To deploy the API Portal with single sign-on:

- An API proxy administrator and an API Portal administrator must configure their API proxy and the API Portal to use their enterprise ID provider for user authentication and management. See Configure SAML Single Sign-On.

To login into API Portal with Broadcom Okta:

After you configure the SAML SSO configuration and get the new IdP activated, you can login into API Portal using Okta.



1. Click **SAML Login** on the API Portal Login screen.
2. Log in using an email address. Broadcom Okta verifies it based on the email domains that were added to the SAML SSO configuration.
   The API Portal dashboard opens after successful login.

Integrate On-Premise API Proxies

Enterprises that deploy the CA API Management solution require an on-premise API proxy and an instance of the API Portal. This article describes how to integrate one or more clusters of on-premise API proxies with API Portal. It also explains how to update the integration software on the API proxy when necessary.

IMPORTANT

For hybrid customers, you must keep API Portal integration software up to date in your solution as the software does not perform this automatically. If you do not, your solution might not take advantage of new features, defect fixes, or security patches.

If your on-premise Gateway requires a proxy setting for outbound traffic or connections, you must modify the Routing Assertions in your specific policies or services.

In this article, learn how to:

After the administrator deploys the CA API Management solution, the following functionality is available on API Portal:

- Publish an API and view the details of the API.
- Create and manage users.
- Self-register to Portal and view the APIs.
- Create organizations and account plans.
- Approve or reject requests from the Requests page.
- Perform configurations from the Settings page.
- View APIs in the API Explorer or Swagger UI.

  NOTE
  You can test APIs from the API Explorer option only if an API proxy is enrolled with Portal.

TIP
When you enroll more than one API cluster with API Portal, you can publish APIs and can manage API keys across multiple environments from a single Portal instance. Examples of multiple environments include: developer, test, production.

After integrating the on-premise API proxy clusters with a Portal instance, users can do the following tasks:

- Publish APIs
- Assign organizations to specific proxies
- Manage API keys
- View the analytics data in the Analytics dashboard
- Test the APIs on a proxy using the API Explorer or Swagger UI

  NOTE
  API Explorer is only accessible through the API Portal/Ingress tenant.

Integrate On-Premise API Proxy Clusters

During API proxy enrollment, you can name a proxy, set the API and application deployment type, and assign organizations to the proxy.

You can set the following API and application deployment types for your proxies:

- **Automatic**: Any change to the API or application is deployed automatically to the proxy. For example, whenever an API or application is created, edited, or deleted. This is the default type.
- **On-demand**: API deployments are triggered on-demand by calling the Deployment API. You can access this API from the Portal APIs link in the navigation menu.
- **Scripted**: API and application deployments are integrated into your existing CI/CD process by leveraging the deployment APIs and invoking them from an API deployment script. The deployment APIs retrieve API or application deployment data and update the API or application deployment status for a proxy to keep API Portal updated.

For more information about how to select the API and application deployment type, see Manage API Deployments and Manage Applications.

A proxy can also have specific organizations assigned to it to allow API or application deployment in multiple organizations. After an organization is assigned to a proxy, a Publisher within the organization can deploy an API they own or manage to that proxy.

For more information, see Organizations Assignment.

Prerequisites for API Proxy Enrollment

- A supported version of the API Gateway as specified in the Compatibility Matrix.
- API Portal supports only the default OTK installation. Do not install it with an instance modifier. Also, the OTK must be installed with JDBC connection name **OAuth**.
- The API proxy can make a secure outbound connection on port 443 to API Portal.

  TIP
  Use cURL or Wget to test the port.
- Ensure that there are no global policies, including message-received, configured on the API proxy. Global policies cannot exist while the Gateway is integrated with Portal.

  WARNING

  - Use the enrollment URL within 24 hours, otherwise it expires. Keep it confidential. Before you use the URL, anyone who knows it can enroll a different API proxy with API Portal.
  - We recommend that you use a proper SSL certificate on your on-premise API proxy. If instead you use a self-signed certificate, for the API proxy to work, the Portal Admin must inform all users to configure their browsers to accept the certificate.
  - The API Portal only supports the default OTK installation. It is not compatible with OTKs that are installed with an instance modifier.

  NOTE
  If you have enabled the assertion **Add HTTP Header Strict-Transport-Security** in the OTK policy **OTK Authorization Server Configuration**, then responses include the **Strict-Transport-Security** header (HSTS). This header restricts browser communication to HSTS only. In hybrid deployments of API Portal, the assertion is enabled by default. In SaaS deployments, the assertion is disabled by default. We recommend disabling the HSTS assertion in your hybrid deployment. For more information about this assertion, see the OAuth Toolkit documentation.

Enroll the On-Premise API Proxy Cluster

Follow these steps:

1. Use API Portal to get the enrollment URL:
   a. Log in to API Portal as a Portal administrator.
   b. From the menu bar, click **Manage**, **Proxies**.
   c. Select **Add Proxy**.
      The Add Proxy Details page appears.
   d. Complete the following fields, and then select **Save & Next**:
      - For **Proxy Name**: Give your proxy cluster a unique name.
      - For **API Deployment Type**: Choose between Automatic, On Demand, or Scripted.
        For more information about API deployment types, see Manage API Deployments.
      - For **Key Deployment Type**: Choose between Automatic, On Demand, or Scripted.
        For more information about application and API key deployment types, see Manage Applications.
   e. On the **Add Proxy Organization Assignment** > **Organizations**: Select organizations that have access to this proxy. For more information about how organization assignment affects API deployment to proxies, see Organizations Assignment.

   f. On the Complete Proxy Enrollment page, select **Select URL** to copy the enrollment URL to the clipboard. Do not close or navigate away from the Complete Proxy Enrollment page.
2. Use the Policy Manager to submit the enrollment URL:
   a. Log in to the API proxy as the API proxy administrator.
   b. On the **Tasks** menu, select **Extensions and Add-Ons**, **Enroll with Portal**. The URL is automatically pasted when using the desktop client version of the Policy Manager.
   c. Select **Apply**.

The enrollment process adds the following items to the API proxy:

- New certificate
- New private key
- New cluster properties
- New encapsulated assertions
- New scheduled tasks (which you can edit, but not remove)
- New folders:
  – API Portal Integration
  – API Portal SSO
  – Portal APIs (This folder is not populated until APIs are deployed to the proxy.)

NOTE
If your on-premise API proxy has the CA Mobile Access Gateway (MAG) components installed, we recommend that you hide the social-media login buttons from Portal users, as described below.

Update Integration Software on the API Proxy

Updating to the latest version of the integration software (also known as the Portal Integration bundle) ensures that API and API key deployment synchronization between the Portal and On-Premise proxies are optimized for reliability and scalability. It also ensures that the API Portal is able to capture and present richer data in the Proxy Details page for analysis and troubleshooting.

When an update for the integration software is available as part of a Portal release, its availability is highlighted in the Release Notes. Portal administrators should coordinate with the API proxy administrator to update the integration software on the API proxy.

See Compatibility Matrix to learn what the latest version of the integration software is and how to identify if you have the latest version installed on your proxy.

WARNING

- The update overwrites any customizations to standard services installed by the Portal integration software, policies, policy templates, or encapsulated assertions. The update does not affect non-standard services, policies, policy templates, or encapsulated assertions. It also does not affect scheduled tasks, or the cached age of APIs and Account Plans (cluster properties).
- This update feature does not update the version of the API proxy. This upgrade feature only upgrades the integration software. For information about general API proxy updates, see Upgrade CA API Gateways in the online documentation for the API Gateway.
- **For customers using API Gateway 10 CR1 and higher:** Download the PortalUpgradeAssertion replacement file to replace the existing server module file when performing your update. For more information, see KB 201757: Upgrade Portal Integration bundle operation fails for API Gateway 10 CR1 and above.

Follow these steps:

1. In the Policy Manager, log in to the API proxy as an administrator.

NOTE
If you upgraded your API Portal prior to your attempt in updating the Portal integration bundle, your API proxy may have disconnected. If this is the case, restart the API proxy before proceeding with the integration bundle update:

```
$ service ssg stop
$ service ssg start
```

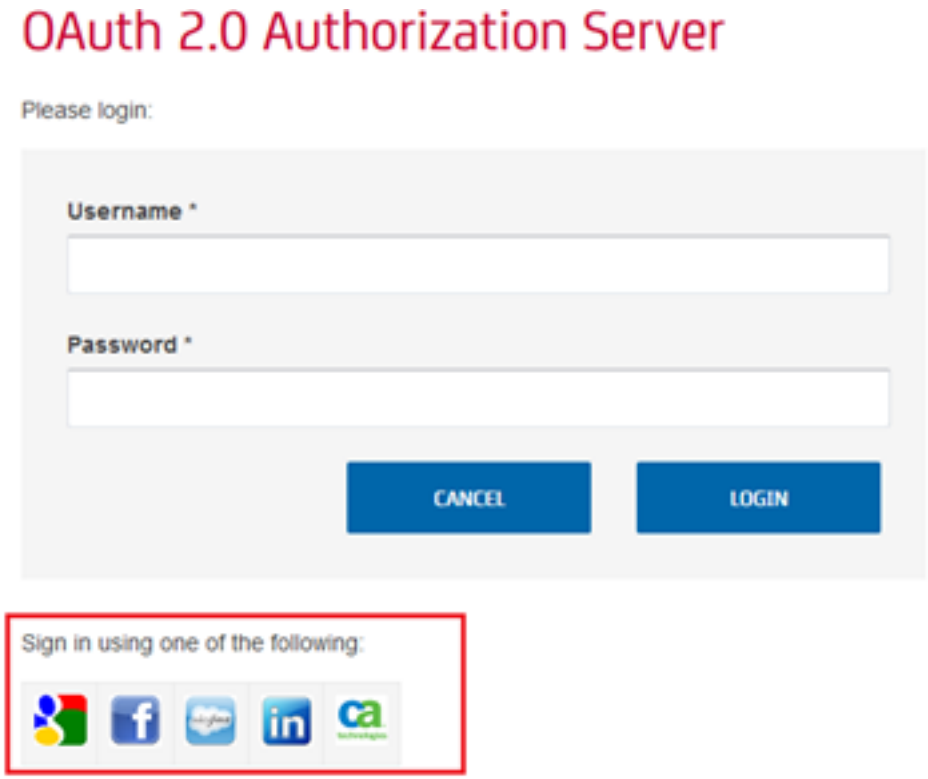
2. (*For API Gateway 10 CR1 and higher only; skip this step if you are using other versions of the Gateway OR if you have already replaced the PortalUpgradeAssertion file after the Portal 5.0.0 upgrade*) Download and replace the PortalUpgradeAssertion file. Follow the instructions in KB 201757: Upgrade Portal Integration bundle operation fails for API Gateway 10 CR1 and above.
3. On the **Tasks** menu, click **Extensions and Add-Ons**, **Update Portal Integration**.
4. Restart the API proxy. To do this, open a privileged shell on the API proxy and then run these commands:

```
$ service ssg stop
$ service ssg start
```

For more information, see 'Using the Privileged Shell' in the online documentation for the API Gateway.

Hide Social-Media Login Buttons from Portal Users

If your on-premise API proxy has the CA Mobile Access Gateway (MAG) components installed, then the OAuth 2.0 Authorization Login dialog displays social-media login buttons to Portal users. However, Portal SaaS does not support social media login. So when a Portal user clicks a social-media login button (shown next) an error message appears.



To hide the social-media login buttons from Portal users, API proxy administrators can edit the "MAG Enabled Social Login Providers" policy fragment.

Follow these steps:

1. Start the Policy Manager.

2. Log in to the proxy as an administrator.
3. Locate the **MAG Enabled Social Login Providers** policy fragment in the MAG Social Login folder: MAG-<version>, configuration, MAG Social Login.
4. Set the following context variables to false:
   enable_google
   enable_facebook
   enable_linkedin
   enable_salesforce
   enable_enterprise
   enable_device2device

Clean Up the API Gateway and Portal after a Failed Enrollment

If you tried to enroll a tenant API Gateway with an API Portal but the enrollment failed, then clean up the API Gateway and Portal before you try again.

> **NOTE**
> You can use the following procedures whether you set up the API Proxy on AWS or on another cloud or network.

Step 1. Clean up the tenant API Gateway:

1. In the Policy Manager, log in to the Gateway as a Gateway administrator.
2. On the **Tasks** menu, select **Certificates, Keys and Secrets** and **Manage Certificates**.
3. Remove the PSSG and DSSG certificates.
   > **NOTE**
   > Do not delete the API Gateway self-signed SSL certificate.
4. On the **Tasks** menu, select **Certificates, Keys and Secrets** and **Manage Private Keys**.
5. Remove the portalman private key.
6. On the **Tasks** menu, select **Global Settings** and **Manage Scheduled Tasks**.
7. Remove all scheduled tasks.
8. On the **Tasks** menu, select **Global Settings** and **Manage Cluster-wide Properties**.
9. Remove all properties that begin with *portal*.

Step 2. Remove the API Gateway from the API Portal:

1. Log in to the API Portal as an API Portal administrator.
2. From the menu bar, click **Manage, Proxies**.
3. On the API Proxy page, find the Gateway. Its state is **Cluster is currently pending enrollment completion**.
4. Select **Delete** next to the Gateway you want to remove.

> **NOTE**
> **More information:**
>
> - Manage Proxies

Manage URLs for API Proxy Enrollment

If your site has multiple clusters of API proxies, as a Portal Admin, you can see which ones have not been enrolled with the API Portal (they are "pending enrollment completion"). If you do not intend to enroll a cluster of API proxies that is pending enrollment URLs with the API Portal, delete it.

> **TIP**
> You can also manage your API proxies by way of the Portal API (PAPI) or use this API in your scripts for managing API proxies.
>
> For more information about the PAPI, see Portal API (PAPI).

View the List of Clusters that are Pending Enrollment URLs

1. From the menu bar, select **Manage**, **Proxies**.
   The API Proxy page appears. A list of clusters of API proxies are displayed.

   NOTE
   If you are using API plans, to enroll a Gateway (by entering the enrollment URL in Policy Manager), the Gateway proxy must be version 9.3.0 or newer.
2. Locate the clusters that show the following message in the **API Proxy URL** column:
   ```
   Cluster is currently pending enrollment completion
   ```

Delete API Proxy Clusters the are Pending Enrollment URLs

1. While logged in to the API Portal, from the API Proxy page, on the Actions menu next to the cluster of API proxies that is pending enrollment URLs that you want to delete, select **Delete**.
2. Confirm the deletion by clicking **Ok**.

The API proxy cluster is deleted.

Configure Authentication Schemes

Authentication schemes determine the identity of users who attempt to access the API Portal resources. Administrators can configure the following Identity Providers with API Portal to authenticate users:

- Microsoft Active Directory (AD)
- Lightweight Directory Access Protocol (LDAP)
- SAML Single Sign On (SAML SSO)

Users who log in to CA API Developer Portal using an external authentication scheme cannot be edited in portal. You can, however, map the developer type users to multiple organizations. To do so, the portal administrator has to edit the authentication scheme of these users and set the authorization type to Portal. For more information, see Map IdP Users to Multiple Organizations. This feature allows the administrator to change only the organization and role mapping; other user details cannot be edited.

From the menu bar, select the gear icon, **Authentication** to add, view, and edit the authentication schemes.

Default Authentication Scheme

API Portal provides a default authentication scheme of type CA APIM to manage users in Portal database. Administrator can only perform the following operations on the default authentication scheme:

- Edit the name of the authentication scheme
- Manage password policies
- Enable or disable the encrypting of passwords to reduce external attacks. By default, the encryption option is disabled.

To enable password encryption

1. Log in to the API Portal as an Administrator.
2. From the menu bar, select the gear icon, **Authentication**.
3. On the **Authentication Schemes** page, click the down arrow in the **Actions** section of the default authentication scheme, and select **Edit**.
4. Click **Advanced Configuration**, select **Enable** option.

Log in to Portal with Configured Authentication Schemes

The configured authentication schemes are listed on the API Portal login page. Select an authentication scheme to log in, or set it as a default one. If you set an authentication scheme as a default one, API Portal renders the authentication scheme's login page to prompt user credentials.

Configure Microsoft Active Directory

Administrator can configure API Management SaaS to support Microsoft Active Directory for user authentication. The Lightweight Directory Access Protocol (LDAP) is used to to perform querying against the Microsoft Active Directory to authenticate users.

Prerequisite

Microsoft Active Directory Server that is populated with users, and roles.

How to Configure Microsoft Active Directory

To configure the Microsoft Active Directory, follow the steps:

1. Log in as an administrator.
2. From the menu bar, select the gear icon, **Authentication**.
3. On the **Authentication Schemes** page, click the **Add Authentication Scheme** button.
4. Provide the following information on the **Add Authentication Scheme** page:
   a. **Providers**: Select an LDAP provider from the available providers, and click Next.
   b. **Basic Details**: Specify the provider name, description, a provider icon, and click Next.
   Note: By default, CA icon is set as the provider icon. Provide a different PNG file to change the icon, and ensure that the file size must not exceed 500 KB.
   c. **Provider Configuration**: Provide the following LDAP server details:

Attribute	Description	Example value
Provider Configuration		
LDAP URL	The fully qualified domain name or IP address with specific port of your LDAP server.	ldaps://10.131.63.81:636
Base Distinguished Name	Base Distinguished Name that is used as the basis for user search.	dc=ca,dc=com
Bind Distinguished Name	The complete Bind Distinguished Name of a user with search permissions in LDAP.	cn=admin,ou=admins,dc=ca,dc=com
Bind Password	Password that is associated with the Bind Distinguished Name.	
Lookup Query		
Start *	Specifies the text string that is the beginning of an LDAP search expression.	(&(cn=
End*	Specifies the text string that is the end of an LDAP search expression.	)(objectClass=*))

Effective Query	Defines the combination of Start string, ID-From-Login, and End string of the LDAP search query. ID-From-Login is the username.	(&(cn= ID-From-Login) (objectClass=*))
Attribute Mapping		
Email	Specifies the email address attribute that is defined for users in your LDAP.	mail
First Name	Specifies the first name attribute that is defined for users in your LDAP.	givenName
Last Name	Specifies the last name attribute that is defined for users in your LDAP.	sn
Login	Specifies user ID attribute that is used for login.	cn
Select Authorization Type		
Portal	Select this authorization type, to manage the organization and role mapping from Portal. This means, Portal administrator can map the Developer user (who has logged in to API Portal, at least once) to multiple organizations by editing the user profile.	N/A
Identity Provider	Select this authorization type, to add the organization and role attributes as provided by this external authentication scheme. **Note:** This option does not allow a Portal administrator to map a Developer user to multiple organizations. If you want to change the authorization type from Identity Provider to Portal after creating the authentication scheme, see **Change Existing Authentication Scheme from "Identity Provider" to "Portal"** section from the Map Existing IdP Users to Multiple Organizations topic.	N/A
When Identity Provider is selected as the Authorization type		
Organization	Specifies the organization attribute that a user is associated with.	o
Role	(When IdP is selected) Specifies the user role attribute that is defined in your LDAP.	title

Map the API Portal user roles to the appropriate roles in your IDP	Specifies the API Portal user roles that are similar to the user roles defined in your LDAP: • Portal Administrator • API Owner • Developer • Org Administrator Configure the group attribute to assign the role to all the users present in a group. If the role attribute value is *memberOf*, ensure to provide the full DN in role mapping. The following sample BaseDN is to map the portaladministrators to a group named "Engineering managers" for the domain ca.com: CN=Engineering managers, CN=users, DC=ca, DC=com	

5. Click **Create** to save the configuration.
   Now, Microsoft Active Directory is configured and the Microsoft Active Directory users can be authenticated in API Management SaaS. API Management SaaS login page now lists the configured providers.
   To set an authentication scheme as a default scheme, select Set as Default option in the Actions section from the Authentication Schemes page. Once the Microsoft Active Directory authentication scheme is your default scheme, API Management SaaS renders this login page to prompt for user credentials.
   Note: To add and manage external users from API Management SaaS, use the Users option in the navigation bar. For information about how to manage users from Portal, see the Get Started - User Types, Roles and Permissions section.
6. If you have configured the authorization type as Portal, any new user who logs in to Portal has only Guest user privileges. To map a Developer to multiple organizations, you need a Portal administrator. Use one of the following methods:
   – Use an IdP Publisher with Portal administrator role.
     a. Create another LDAP authentication scheme with authorization type as "Identity Provider".
     b. Add the role as Portal Administrator.
     c. Log in to API Portal as the Portal administrator.
     d. Edit the Developer user profile to map to multiple organizations.
   – Use the Portal administrator added and managed in API Portal. Edit the Developer user profile to map to multiple organizations.

Edit and Delete Microsoft Active Directory Configuration

If your Microsoft Active Directory configuration changes, update the same in API Management SaaS.

To edit the Microsoft Active Directory details, follow the steps:

1. Log in to CA APIM Portal as an Administrator.
2. From the menu bar, select the gear icon, **Authentication**.
3. On the **Authentication Schemes** page, click the down arrow in the **Actions** section of a configured LDAP, and select Edit.
4. In the Edit Authentication Scheme page, select an LDAP configuration to edit. For example, to edit the provider details, select the Provider Configuration option. Make the required changes and click Save.
5. To delete Microsoft Active Directory that is configured with API Management SaaS: On the Authentication Schemes page, click the down arrow in the **Actions** section of a configured LDAP, and select Delete.

Configure Lightweight Directory Access Protocol

Administrator can configure API Management SaaS to support LDAP for user authentication. Other than organization and role, you cannot edit other details for an external IdP user.

Prerequisite: LDAP servers that are populated with users, and roles.

Follow these steps:

1. Log in as an administrator.
2. From the menu bar, select the gear icon, **Authentication**.
3. On the **Authentication Schemes** page, click the **Add Authentication Scheme** button.
4. Provide the following information on the **Add Authentication Scheme** page:
   a. **Providers**: Select an LDAP provider from the available providers, and click Next.
   b. **Basic Details**: Specify the LDAP provider name, description, a provider icon, and click Next.

      NOTE

      By default, CA icon is set as the provider icon. Provide a different PNG file to change the icon, and ensure that the file size must not exceed 500 KB.
   c. **Provider Configuration**: Provide the following LDAP server details:

Attribute	Description	Example value
Connection Details		
LDAP Host	Host name of your LDAP server.	
LDAP Port	Specific port of your LDAP server.	
SSL Enabled?	Select Yes if the connection from the LDAP client to the LDAP server is secure.	
Directory Details		
Base Distinguished Name	Base Distinguished Name that is used as the basis for user search.	dc=ca,dc=com
Bind Distinguished Name	The complete Bind Distinguished Name of a user with search permissions in LDAP.	cn=admin,ou=admins,dc=ca,dc=com
Bind Password	Password that is associated with the Bind Distinguished Name.	
Lookup Query		
Start *	Specifies the text string that is the beginning of an LDAP search expression.	(&(cn=
End *	Specifies the text string that is the end of an LDAP search expression.	)(objectClass=*))
Effective Query	Defines the combination of Start string, ID-From-Login, and End string of the LDAP search query. ID-From-Login is the username.	(&(cn= ID-From-Login) (objectClass=*))
Attribute Mapping		

Email	Specifies the email address attribute that is defined for users in your LDAP.	mail
First Name	Specifies the first name attribute that is defined for users in your LDAP.	givenName
Last Name	Specifies the last name attribute that is defined for users in your LDAP.	sn
Select Authorization Type		
Portal	Select this authorization type, to manage the organization and role mapping from Portal. This means, you can map the Developer user (who has logged in to API Portal, at least once) to multiple organizations by editing the user profile.	N/A
Identity Provider	Select this authorization type, to add the organization and role attributes as provided by this external authentication scheme. **Note:** This option does not allow a Portal administrator to map a Developer user to multiple organizations. If you want to change the authorization type from Identity Provider to Portal after creating the authentication scheme, see **Change Existing Authentication Scheme from "Identity Provider" to "Portal"** section from the Map IdP Users to Multiple Organizations topic.	N/A
When Identity Provider is selected as the Authorization type		
Organization	Specifies the organization attribute that a user is associated with.	o
Role	Specifies the user role attribute that is defined in your LDAP.	title

Map the API Portal user roles to the appropriate roles in your IDP	Specifies the API Portal user roles that are similar to the user roles defined in your LDAP: • Portal Administrator • API Owner • Developer • Org Administrator Configure the group attribute to assign the role to all the users present in a group. If the role attribute value is *memberOf*, ensure to provide the full DN in role mapping. The following sample BaseDN is to map the portaladministrators to a group named "Engineering managers" for the domain ca.com: CN=Engineering managers, CN=users, DC=ca, DC=com	

5. Click Create to save the LDAP configuration.
   Now, LDAP is configured and the LDAP users can be authenticated in CA APIM Portal. CA APIM Portal login page now lists the configured LDAP providers.
   To set an authentication scheme as a default scheme, select Set as Default option in the Actions section from the Authentication Schemes page. Once the LDAP authentication scheme is your default scheme, CA APIM Portal renders this LDAP login page to prompt for user credentials.
   Note: To add and manage external users from API Portal, use the Users option in the navigation bar. For information about how to manage users from Portal, see the Get Started - User Types, Roles and Permissions section.
6. If you have configured the authorization type as Portal, any new user who logs in to Portal has only Guest user privileges. To map a Developer to multiple organizations, you need a Portal administrator. Use one of the following methods:
   – Use an IdP Publisher with Portal administrator role.
     a. Create another LDAP authentication scheme with authorization type as "Identity Provider".
     b. Add the role as Portal Administrator.
     c. Log in to API Portal as the Portal administrator.
     d. Edit the Developer user profile to map to multiple organizations.
   – Use the Portal administrator added and managed in API Portal. Edit the Developer user profile to map to multiple organizations.

Edit and Delete LDAP Configuration

If your LDAP configuration changes, update the same in CA APIM Portal.

To edit the LDAP details, follow the steps:

1. Log in to CA APIM Portal as an Administrator.
2. From the menu bar, select the gear icon, **Authentication**.
3. On the **Authentication Schemes** page, click the down arrow in the **Actions** section of a configured LDAP, and select Edit.
4. In the Edit Authentication Scheme page, select an LDAP configuration to edit. For example, to edit the provider details, select the Provider Configuration option. Make the required changes and click Save.

   NOTE

   You need to enter the Bind Password in order to save your changes.

5. To delete LDAP that is configured with CAPIM Portal: On the Authentication Schemes page, click the down arrow in the **Actions** section of a configured LDAP, and select Delete.

Configure SSO for Local SaaS Environments

This article describes how to configure an IdP in your local environment for testing SAML SSO features.

SAML 2.0 is an XML-based protocol that uses security tokens to pass user authentication and authorization data between an IdP, and a service provider. API Management SaaS adheres to SAML 2.0 standards and uses user authentication when integrated with a SAML IdP system. Employing SAML IdP to authenticate and manage API Portal users provides the benefit of SSO.

In the SAML context, the API Management SaaS is the service provider (SP).

The following tasks are supported:

- Configuration of multiple SAML SSO schemes on CA API Developer Portal
- Service provider initiated Web Single Sign-On (Web SSO).

Note the following conditions:

- If you are using the Portal API, SAML SSO is not supported. The Portal API login policy does not support a third-party system for authentication.
- During SSO testing, the Gateway serves as the IdP.
- This procedure applies to SaaS and hybrid deployments.

This article contains the following information:

Prerequisites

Verify that your environment meets the following prerequisites before you configure an IdP:

- CA API Management SaaS is running.
- The Portal Gateway and the Tenant Gateways are running.
- SoapUI or your favorite client tool is ready to consume services.

Integrate SAML SSO

Administrators can integrate SAML SSO using any one of following ways:

- Integrate SAML SSO from the Tenant Gateway
- Integrate SAML SSO from API Management SaaS

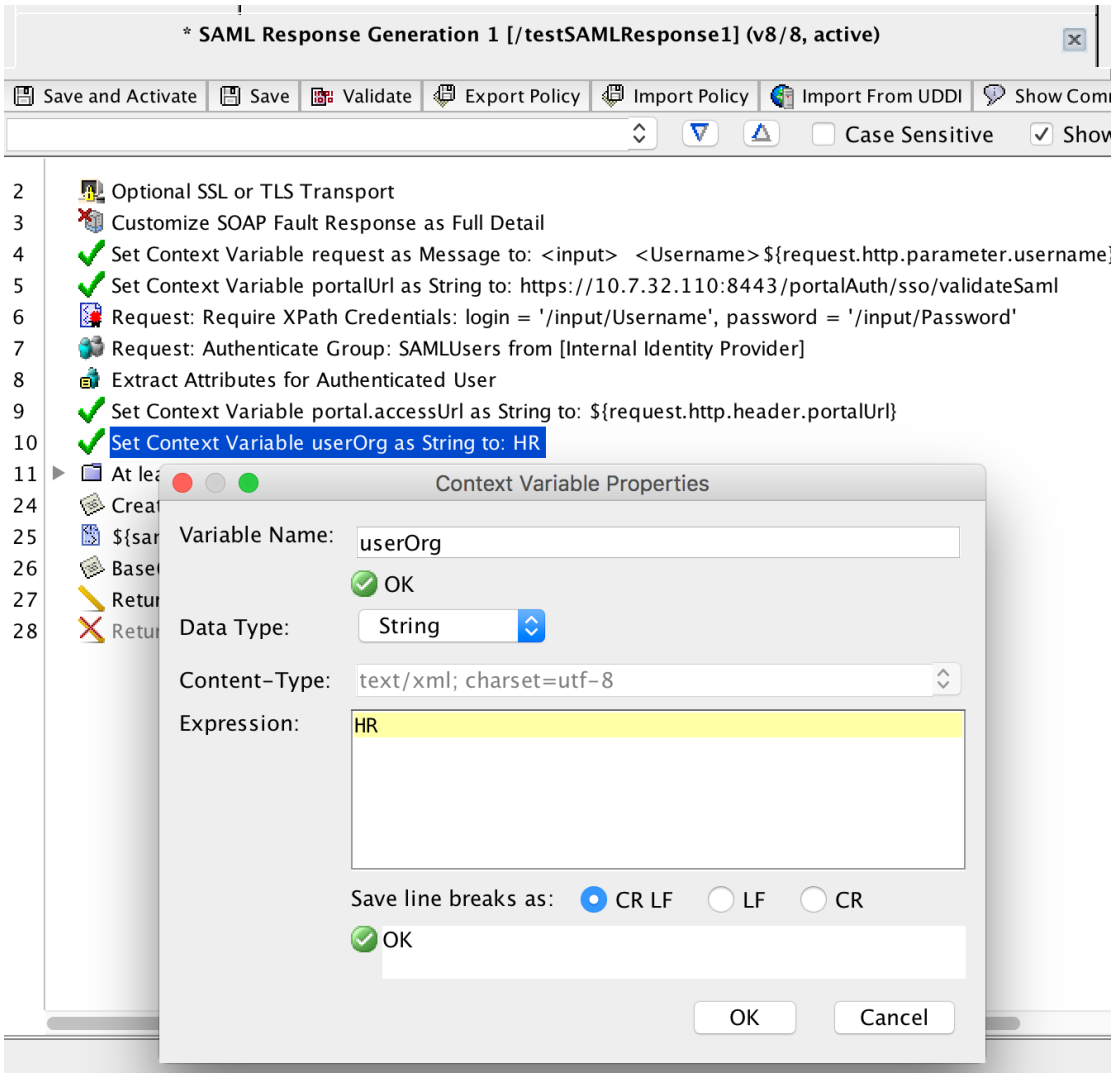
Integrate SAML SSO from the Tenant Gateway

Complete the following steps to create mock SAML IdP services in the Tenant Gateway for testing SSO:

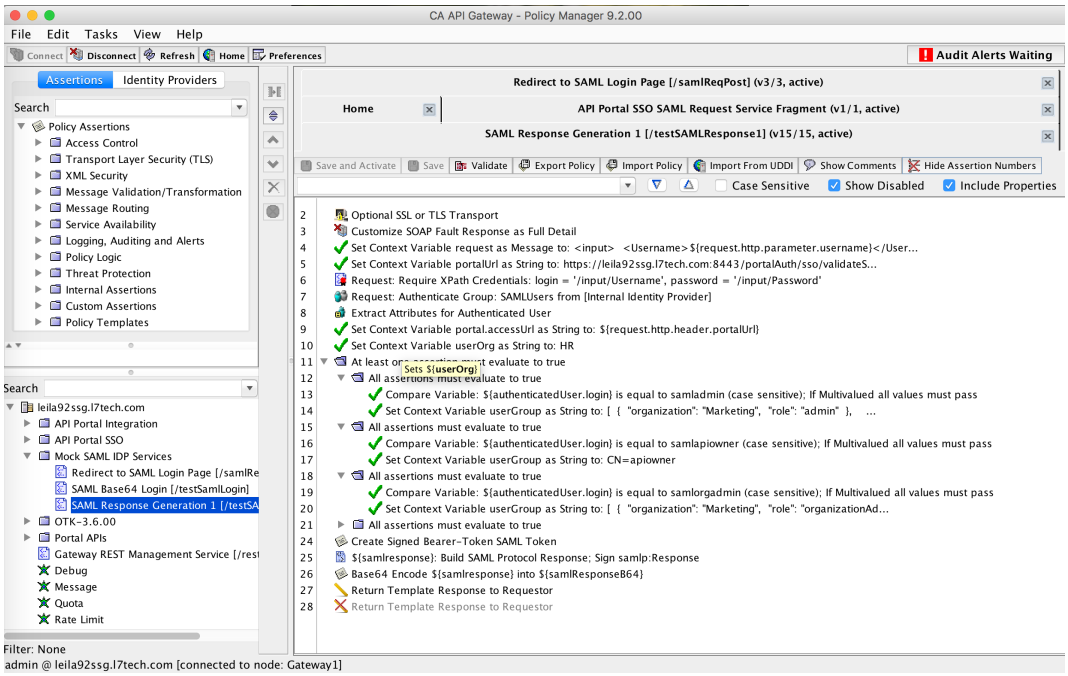
1. Use Chrome Postman, or another tool, such as DHC, to send a message to Tenant Gateway endpoint:
   - **Endpoint**: "`https://<Your_Tenant_Gateway>:9443/restman/1.0/bundle`"
   - **Method**: PUT
   - **Authorization**: Basic Auth: your Tenant Gateway Policy Manager login credential. (for example, pmadmin / 7layer)
   - **Content-type**: application/xml
   - You do not need Origin or Referrer headers
   - **Body**: select **binary**, select **mock_sso_bundle.xml**
     If consumption succeeds, you are expected to get a 200 OK response.
2. Log in to the Policy Manager and perform the following steps:

- Verify that a folder that is named **Mock SAML IDP Services** appears under the root tree.
  Note: If you were logged in to Policy Manager before you sent the request, manually refresh the Policy Manager to see the folder.
  Three services exist under the folder:
  - Redirect to SAML Login Page [/samlReqPost]
  - SAML Base64 Login [/testSamlLogin]
  - SAML Response Generation 1 [/testSAMLResponse]
- Verify that the user group **SAMLUsers** exists in the Gateway, as follows:
  a. Navigate to **Home**, **Search Identity Provider**, **Type: Groups**
  b. Select **Search**. The SAMLUsers group appears.

3. In **Mock SAML IDP Services**, replace the Gateway URLs with URLs for your environment:
   - In line 3, Redirect to SAML Login Page [/samlReqPost], set Property/Header Value to
     `http://<Your_Tenant_Gateway>:8080/testSamlLogin`
     For example, Property/Header Value: "http://emportal-tssg1.abc.com:8080/testSamlLogin"
   - In line 5, SAML Response Generation 1 [/testSAMLResponse1], Set Context Variable portalUrl as String to:
     `https://<Your_Tenant_Gateway>:8443/portalAuth/sso/validateSaml`
     Note: The URL should match the SSO endpoint on the API Portal SSO Configuration page.
4. Create users for different Portal roles and assign them to the **SAMLUsers** group, as follows:
   a. Go to **Home**, **Create Internal User**. Create a user called samladmin. Specify the following information:
      Password (Specify any password)
      First Name
      Last Name
      Email
   b. Repeat step a to create the following additional users:
      samlapiowner
      samlorgadmin
      samldev
   c. Go to **Home**, **Search Identity Provider**, **Type: Users**. Search for and select samladmin. On the Groups tab, add the SAMLUsers group to this user.
   d. Repeat step c for the following users:
      samlapiowner
      samlorgadmin
      samldev
5. In line 10, set context variable userOrg to the default organization that is used in userGroup role mapping payload:



6. Create different Portal roles in "SAML Response Generation 1 [/testSAMLResponse1]" as follows:



The preceding image illustrates the following changes:

Set ${authenticatedUser.login} to be "samladmin", "samlapiowner", "samlorgadmin", and "samlorgdev".

Set userGroup to "CN=admin", "CN=apiowner", "CN=devorgadministrators", "CN=developer".
The following example shows how to configure userGroup to support different roles in different organizations:

```
[
 {
 "organization": "HR",
 "role": "developer"
 },
 {
 "organization": "Service",
 "role": "developer"
 },
 {
 "organization": "Marketing",
 "role": "organizationAdmin"
 }
]
```

Integrate SAML SSO From API Management SaaS

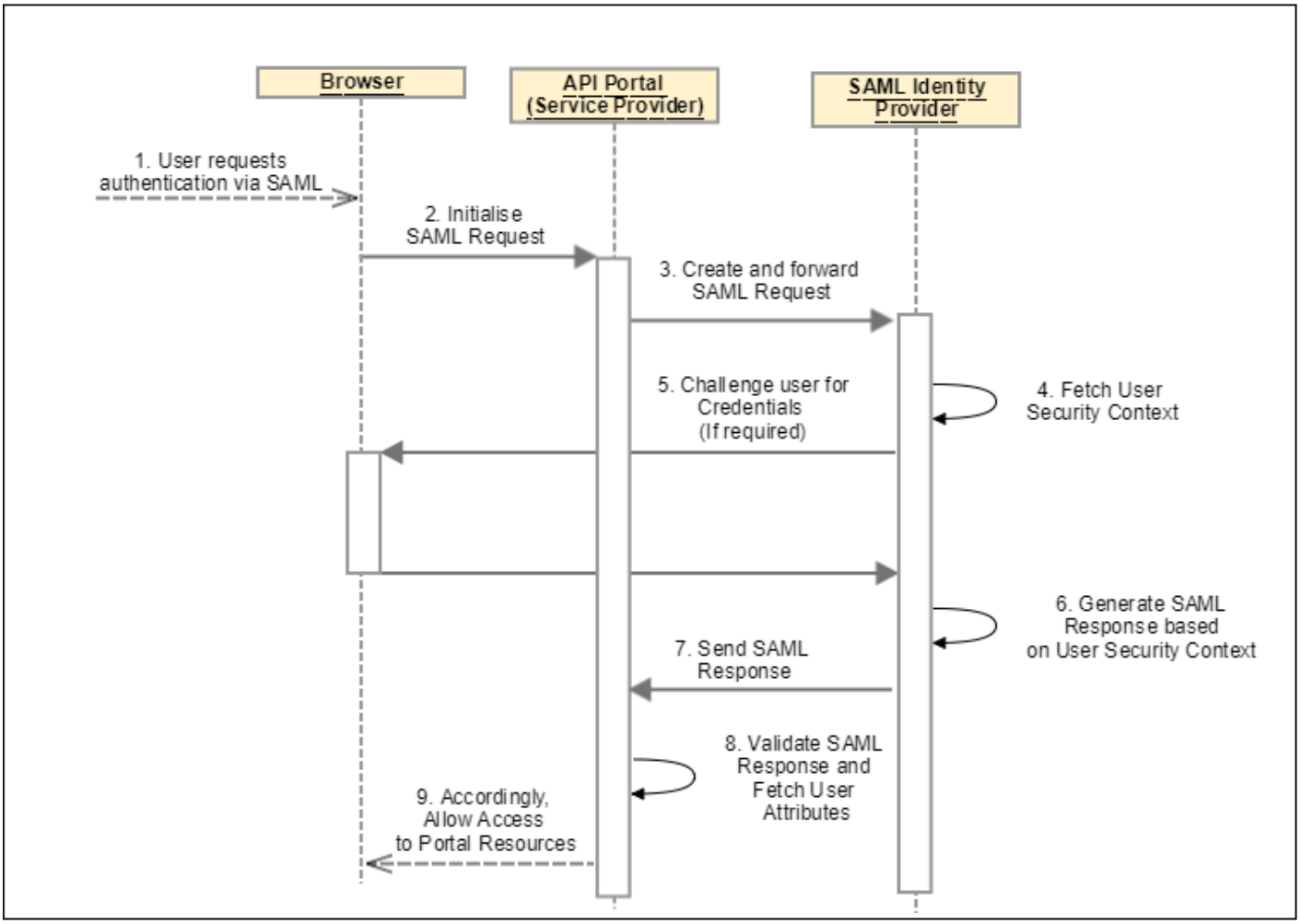
The following tasks are related to creating and managing a SAML SSO configuration:

> **NOTE**
>
> **More Information:** FAQ.

SAML Authentication Workflow

The following sequence diagram shows the SAML authentication workflow in CA API Developer API Portal.

Figure 1: SAML Authentication flow



Create a SAML SSO Authentication Scheme

You create the authentication scheme by adding provider configuration values, then mapping user attributes and roles. The resulting authentication scheme can be set as the default to render SAML login page.

Follow these steps:

1. Log in as an administrator.
2. Select **Administration**, **Authentication**.
3. On the **Authentication Schemes** page, select the **Add Authentication Scheme** button.
4. For **Providers**, select **SAML SSO** provider from the available providers, and select **Next**.
5. For **Basic Details**, type the SAML SSO provider name and a description.
6. (Optional) Add a provider icon, and select **Next**. The provider icon must be a PNG file, and the size must not exceed 500 KB.
7. Add provider configuration values, then map user attributes and roles.
   See sections following these instructions for details.
8. Select **Create** to save the SAML SSO configuration.

SAML authentication scheme is configured.

Set SAML Authentication Scheme as a Default Scheme

After API Portal is integrated with SAML IdP, you can set the SAML authentication scheme as a default scheme. On the Authentication Schemes page, for a SAML authentication scheme select **Set as Default** in the **Actions** menu. Once the SAML SSO authentication scheme is your default scheme, API Portal renders the selected SAML IdP login page to prompt for user credentials.

If the SAML authentication scheme is not set as a default authentication scheme, the SAML Provider is listed on the API Portal login page. Select the SAML Provider to open the SAML IdP login page. Provide the user credentials that are verified on the SAML IdP, and the user is logged in to CA API Developer Portal.

If the SAML Provider is set as default and you are unable to log in using SAML, use the *hostname/admin/login* URL to log in to API Portal and verify the SAML provider configuration.

NOTE

CA API Developer portal does not support user creation and management in IdP. User management has to be done at the SAML IdP .

Having configured IdP with Portal, Portal administrators and Organization administrators can still create and manage users in Portal authenticated using CA APIM Authentication Scheme. For information about how to manage users from Portal, see the User Types, Roles and Permissions section.

For solutions to troubleshoot issues that may occur while configuring the SAML authentication schemes, see the FAQ, sections for queries about the SAML SSO integration with API Portal.

For information about how to set up SSO for the API Gateway, see "Working with CA Single Sign-On" in the API Gateway documentation.

#unique_82

Add Provider Configuration Details

Fill in provider configuration details as shown in the following table.

Attribute	Description	Notes
Assertion Consumer Service (ACS) URL	Assertion Consumer Service (ACS) URL for API Portal Authentication. SAML response is received at this URL. The field value is populated and is non-editable.	
Identity Provider URL	SAML Identity Provider URL for user authentication.	For example, if the IdP is Salesforce: http://mydomain.my.salesforce.com?login. The URL is the SSO login page for the API Portal.
SAML Binding	Select the SAML Binding to determine how SAML requests map to communication protocols. Specify the request in POST or Redirect form to send it to the SAML IdP.	The value is read-only. No configuration available.
SAML Token Attribute	The value is populated with the SAML Token attribute name that contains the user information.	
SAML Token Attributeln	Defines how the SAML Token Attribute content is returned from the SAML IdP. The content is returned as a parameter.	The value is read-only. No configuration available.

Service provider ID	Specify the service provider identification that identifies the API Management SaaS service to establish the connection between IdP and the Service provider.	If you do not have any specific service provider ID, use the default ID that API Management SaaS generates.
Issuer ID	Specify the SAML issuer ID.	The SAML Response issuer should be set as the IdP's entity ID.
Upload Trusted Certificate.	Upload a trusted certificate in X.509 format to validate the signed SAML response that an Identity Provider provides.	

Map User Attributes and Roles

Map API Portal user attributes to conceptually similar attributes that the SAML IdP returns.

The following attribute mappings are required:

User Attribute	Notes
Email	Specifies the email address attribute that is defined for users in your Identity Provider.
First Name	Specifies the first name attribute that is defined for users in your Identity Provider.
Last Name	Specifies the last name attribute that is defined for users in your Identity Provider.
Login	Specifies user ID attribute that is used for login.
Organization	Specifies the organization attribute that a user is associated with.
Role	Specifies the user role attribute that is defined in your identity provider. Select a role from the available list and map it to conceptually similar user roles in your SAML IdP: 1. a. • Portal Administrator • API Owner • Developer • Org Administrator For more information about the roles and responsibilities of the API Portal users, see the User Types, Roles and Permissions section.

Establish Trust on SAML IdP

Collect the information that is required to establish trust from the Provider Configuration table. Ensure that the ACS URL provided is used to establish the trust.

The following values are required to establish trust on SAML IdP:

Information Type	Required Values
Service provider-specific information.	Requires the following values: • Assertion Consumer Service (ACS) URL URL where the SAML response is received from the IdP. • Service provider ID API Portal entity ID, or SAML request issuer. If the IdP does not have a service provider ID, use the default value that API Portal displays in the configuration screen.
API Portal-specific information:	Requires the following values: • SAML Token Attribute • SAML Token Attributeln

Edit SAML SSO Configuration

To edit the SAML SSO details:

1. Log in to the API Portal as an Administrator.
2. Select **Administration**, **Authentication**.
3. On the **Authentication Schemes** page, select the down arrow in the **Actions** section of a configured SAML SSO, and select **Edit**.
4. In the Edit Authentication Scheme page, select SAML SSO configuration to edit. For example, to edit the provider details, select the Provider Configuration option. Make the required changes and select **Save**.

Delete SAML SSO Configuration

To delete the SAML SSO configuration:

1. Log in to the API Portal as an Administrator.
2. Select **Administration**, **Authentication**.

On the Authentication Schemes page, select the down arrow in the **Actions** section of a configured SAML SSO, and select **Delete**.

Troubleshoot

Issues that occur while Configuring the SAML Authentication Schemes

This section describes the solutions to troubleshoot issues that may occur while configuring the SAML authentication schemes.

Symptom:

Creating the SAML authentication scheme on API Portal throws the following error:

The specified username and password was invalid.

Reason:
The issue may be due to one of the following reasons:

- incorrect Identity Provider URL, or Issuer ID, or trusted certificate is provided as the provider configuration details.
- incorrect Assertion Consumer Service (ACS) URL, or Service provide ID is provided while establishing the trust on IdP.
- incorrect mapping of the Role or Organization attributes.

Solution:

Ensure the:

- provider configuration details are valid.
- service provider ID and ACS URL are similar to the one that exists on API Portal.
- role attribute that is mapped on API Portal is conceptually similar in your SAML IdP. The role attribute mapping that is returned in the SAML response should contain one of the roles that are mapped on API Portal as role attributes.
- organization that SAML response returns as part of organization attribute mapping must exist in API Portal.

If the issue persists after you have ensured all the values for creating authentication schemes are correct, we recommend re-creating the authentication scheme.

Issues that Occur while Configuring SSO for Local SaaS Environments

API Portal SSO Fragments are Not Updated

Issue: In existing environments, you do not see updated Fragments in the API Portal SSO folder on the Tenant Gateway after running updaterSA.

Solution: Follow these steps:

1. Log in to the Tenant Gateway Policy Manager.
2. Go to **Tasks**, **Manage Scheduled Tasks**.
3. Remove **Portal Sync SSO Configuration**.
4. Delete the API Portal SSO folder under the root tree.
5. Open the following URL in a browser:
   https://<*Portal Gateway Host Name*>:9446/enroll/tenant1?reset=true
6. Log in as the admin user.
7. On the Enroll API Proxy page, select **SELECT URL** to copy the URL.
8. Log in to the Tenant Gateway Policy Manager.
9. Go to **Tasks**, **Additional Actions**, **Enroll with Portal**, Paste the Enrollment URL here, and select **Apply**. Select **OK** on the confirmation screen.
10. Select OK again.
11. Refresh the Tenant Gateway Policy Manager.
12. Verify the following steps:
    - The **API Portal SSO** folder was recreated and contains updated Fragments.
    - The Portal Sync SSO Configuration appears under Scheduled Tasks.
    - You can successfully log in to the API Portal with saml-users.

Open Listen Port 9448 on Portal Gateway for Existing Environments

Issue: The Portal and Tenant Gateways do not sync for SSO in existing environments.

Solution: Manually open the listen port 9448 on the Portal Gateway.

Follow these steps:

1. Use SoapUI or other client tools to send a PUT message to your Portal Gateway:
   - Endpoint: "https://<*Your_Portal_Gateway*>:8443/restman/1.0/listenPorts/41ff0f5f2b43e41d8fb814cf99cda2c5"
   - Method: PUT
   - Request HEADER: Basic authorization: Specify the Portal Policy Manager login credentials. (for example, pmadmin / 7layer)
   - Request BODY: Content-Type: application\xml; Body:

```
<l7:ListenPort id="41ff0f5f2b43e41d8fb814cf99cda2c5" version="2"
xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
```

```
<l7:Name>TSSG Sync SSO Port</l7:Name>
<l7:Enabled>true</l7:Enabled>
<l7:Protocol>HTTPS</l7:Protocol>
<l7:Port>9448</l7:Port>
<l7:EnabledFeatures>
<l7:StringValue>Published service message input</l7:StringValue>
</l7:EnabledFeatures>
<l7:TargetServiceReference id="22a6e8737c85fc3833fb7164f029efcf"
 resourceUri="http://ns.l7tech.com/2010/04/gateway-management/services"/>
<l7:TlsSettings>
<l7:ClientAuthentication>Optional</l7:ClientAuthentication>
<l7:EnabledVersions>
<l7:StringValue>TLSv1.1</l7:StringValue>
<l7:StringValue>TLSv1.2</l7:StringValue>
</l7:EnabledVersions>
<l7:EnabledCipherSuites>
<l7:StringValue>TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384</l7:StringValue>
<l7:StringValue>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384</l7:StringValue>
<l7:StringValue>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA</l7:StringValue>
<l7:StringValue>TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384</l7:StringValue>
<l7:StringValue>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384</l7:StringValue>
<l7:StringValue>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA</l7:StringValue>
<l7:StringValue>TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256</l7:StringValue>
<l7:StringValue>TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256</l7:StringValue>
<l7:StringValue>TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA</l7:StringValue>
<l7:StringValue>TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256</l7:StringValue>
<l7:StringValue>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256</l7:StringValue>
<l7:StringValue>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA</l7:StringValue>
<l7:StringValue>TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA</l7:StringValue>
<l7:StringValue>TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA</l7:StringValue>
<l7:StringValue>TLS_ECDHE_ECDSA_WITH_RC4_128_SHA</l7:StringValue>
<l7:StringValue>TLS_ECDHE_RSA_WITH_RC4_128_SHA</l7:StringValue>
<l7:StringValue>TLS_DHE_RSA_WITH_AES_256_GCM_SHA384</l7:StringValue>
<l7:StringValue>TLS_DHE_RSA_WITH_AES_256_CBC_SHA</l7:StringValue>
<l7:StringValue>TLS_RSA_WITH_AES_256_GCM_SHA384</l7:StringValue>
<l7:StringValue>TLS_RSA_WITH_AES_256_CBC_SHA</l7:StringValue>
<l7:StringValue>TLS_DHE_RSA_WITH_AES_128_GCM_SHA256</l7:StringValue>
<l7:StringValue>TLS_DHE_RSA_WITH_AES_128_CBC_SHA</l7:StringValue>
<l7:StringValue>TLS_RSA_WITH_AES_128_GCM_SHA256</l7:StringValue>
<l7:StringValue>TLS_RSA_WITH_AES_128_CBC_SHA</l7:StringValue>
<l7:StringValue>SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA</l7:StringValue>
<l7:StringValue>SSL_RSA_WITH_3DES_EDE_CBC_SHA</l7:StringValue>
<l7:StringValue>SSL_RSA_WITH_RC4_128_SHA</l7:StringValue>
<l7:StringValue>SSL_RSA_WITH_RC4_128_MD5</l7:StringValue>
<l7:StringValue>SSL_DHE_RSA_WITH_DES_CBC_SHA</l7:StringValue>
<l7:StringValue>SSL_RSA_WITH_DES_CBC_SHA</l7:StringValue>
```

```
<l7:StringValue>TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA</l7:StringValue>
<l7:StringValue>TLS_ECDH_RSA_WITH_AES_256_CBC_SHA</l7:StringValue>
<l7:StringValue>TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA</l7:StringValue>
<l7:StringValue>TLS_ECDH_RSA_WITH_AES_128_CBC_SHA</l7:StringValue>
<l7:StringValue>TLS_ECDH_ECDSA_WITH_RC4_128_SHA</l7:StringValue>
<l7:StringValue>TLS_ECDH_RSA_WITH_RC4_128_SHA</l7:StringValue>
<l7:StringValue>TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA</l7:StringValue>
<l7:StringValue>TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA</l7:StringValue>
</l7:EnabledCipherSuites>
</l7:TlsSettings>
<l7:Properties>
<l7:Property key="useExtendedFtpCommandSet">
<l7:StringValue>false</l7:StringValue>
</l7:Property>
</l7:Properties>
</l7:ListenPort>
```

2. If the response succeeds, you see the " 201 Created" response.
3. Log in to the Portal Policy Manager.
4. Go to **Tasks**, **Manage Listen Ports**.
5. Verify that port 9448 is open for the TSSG Sync SSO Port.

Configure SAML Single Sign-On

SAML 2.0 is an XML-based protocol that uses security tokens to pass user authentication and authorization data between an IdP, and a service provider. API Management SaaS adheres to SAML 2.0 standards and uses user authentication when integrated with a SAML IdP system. Employing SAML IdP to authenticate and manage API Portal users provides the benefit of SSO.

In the SAML context, the API Management SaaS is the service provider (SP).

The following tasks are supported:

- Configuration of multiple SAML SSO schemes on CA API Developer Portal
- Service provider initiated Web Single Sign-On (Web SSO).

> **NOTE**
>
> To log in to portal, SAML SSO users need to use their IdP UI.

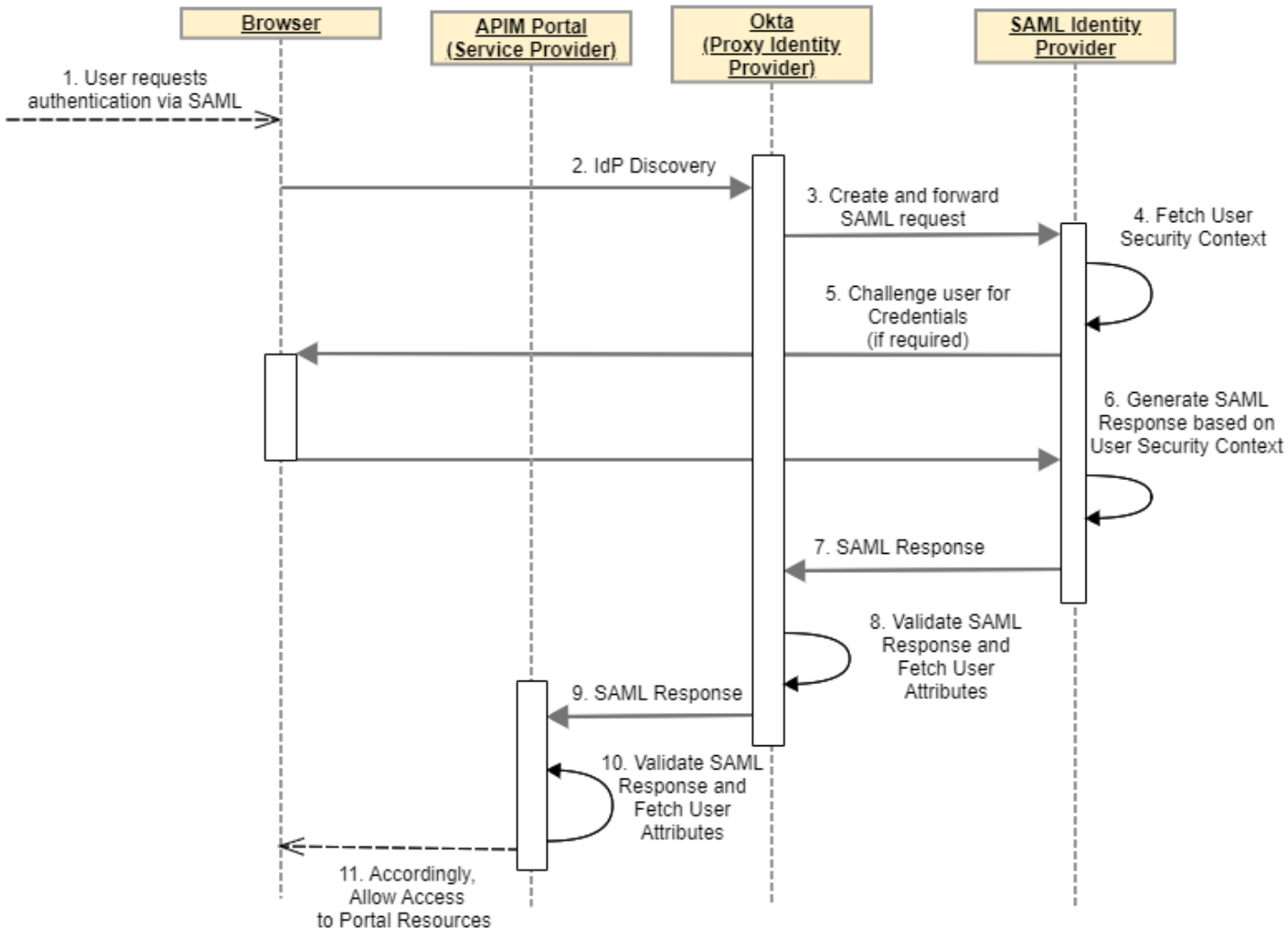
The following tasks are related to creating and managing a SAML SSO configuration:

> **NOTE**
>
> **More Information:** FAQs.

SAML Authentication Workflow

The following sequence diagram shows the SAML authentication workflow in API Developer API Portal.

Figure 2: SAML Authentication Workflow in API Portal



Create a SAML SSO Authentication Scheme

You create the authentication scheme by adding provider configuration values, then mapping user attributes and roles. The resulting authentication scheme can be set as the default to render SAML login page.

Follow these steps:

1. Log in as an administrator.
2. From the menu bar, select the gear icon, **Authentication**.
3. On the **Authentication Schemes** page, select the **Add Authentication Scheme** button.
4. For **Providers**, select **SAML SSO (new)** provider from the available providers, and select **Next**.
5. For **Basic Details**, type the SAML SSO provider name and a description.
6. (Optional) Add a provider icon, and select **Next**. The provider icon must be a PNG file, and the size must not exceed 500 KB.
7. Add Identity Provider configuration values as shown below:

Attribute	Description	Notes
Identity Provider URL	SAML Identity Provider URL for user authentication.	For example, if the IdP is Salesforce: http://mydomain.my.salesforce.com?login. The URL is the SSO login page for the API Portal.
SAML Binding	Select the SAML Binding to determine how SAML requests map to communication protocols. Specify the request in POST or Redirect form to send it to the SAML IdP.	The value is read-only. No configuration available.
SAML Token Attribute	The value is populated with the SAML Token attribute name that contains the user information.	
SAML Token AttributeIn	Defines how the SAML Token Attribute content is returned from the SAML IdP. The content is returned as a parameter.	The value is read-only. No configuration available.
Service provider ID	Specify the service provider identification that identifies the API Management SaaS service to establish the connection between IdP and the Service provider.	If you do not have any specific service provider ID, use the default ID that API Management SaaS generates.
Issuer ID	Specify the SAML issuer ID.	The SAML Response issuer should be set as the IdP's entity ID.
Upload Trusted Certificate.	Upload a trusted certificate in X.509 format to validate the signed SAML response that an Identity Provider provides.	

8. Map API Portal user attributes to conceptually similar attributes that the SAML IdP returns. The following attribute mappings are required:

User Attribute	**Notes**
Email	Specifies the email address attribute that is defined for users in your Identity Pro
First Name	Specifies the first name attribute that is defined for users in your Identity Provide
Last Name	Specifies the last name attribute that is defined for users in your Identity Provide
Login	Specifies user ID attribute that is used for login.
Add Email Domains	Specify valid unique email domains of the users in this IdP. Based on this email comma-separated values or use the **Add** button to add email domains individua
Select Authorization Type	
Portal	Select this authorization type, to manage the organization and role mapping fron Portal, at least once) to multiple organizations by editing the user profile.
Identity Provider	Select this authorization type, to add the organization and role attributes as prov **Note:** This option does not allow a Portal administrator to map a Developer user Identity Provider to Portal after creating the authentication scheme, see **Change** from the Map Existing IdP Users to Multiple Organizations topic.
When Identity Provider is selected as the Authorization type	
Organization	Specifies the organization attribute that a user is associated with.

Role	Specifies the user role attribute that is defined in your identity provider. Select a role from the available list and map it to conceptually similar user roles • Portal Administrator • API Owner • Developer • Org Administrator For more information about the roles and responsibilities of the API Portal users

9. Select **Create** to save the SAML SSO configuration.
   The Authentication Schemes page opens showing the new SAML SSO configuration in the list. It also shows the Assertion Consumer Service (ACS) URL for the configuration.

   ATTENTION
   After you create the SAML SSO configuration, the new IdP will be in **Inactive** state. Contact Layer7 Support and the team will interface with Broadcom IT to review and activate the IdP. You can then start using the new IDP to login into API Portal.

   SAML authentication scheme is configured.

   NOTE
   If you have configured the authorization type as Portal, any new user who logs in to Portal has only Guest user privileges.

Map a Developer to Multiple Organizations

To map a Developer to multiple organizations, you need a Portal administrator. Use one of the following methods:

- Use an IdP user with Portal administrator role.
  a. Create another SAML SSO authentication scheme with authorization type as "Identity Provider".
  b. Add the role as Portal Administrator.
  c. Log in to API Portal as the Portal administrator.
  d. Edit the Developer user profile to map to multiple organizations.
- Use the default Portal administrator added and managed in API Portal.
  – Edit the Developer user profile to map to multiple organizations.

Set SAML Authentication Scheme as a Default Scheme

IMPORTANT
(Only for SaaS users) You can only configure the default login approach to be "SAML" and cannot choose the default IdP within the SAML.

After API Portal is integrated with SAML IdP, you can set the SAML authentication scheme as a default scheme. On the Authentication Schemes page, for a SAML authentication scheme select **Set as Default** in the **Actions** menu. Once the SAML SSO authentication scheme is your default scheme, API Portal renders the selected SAML IdP login page to prompt for user credentials.

If the SAML authentication scheme is not set as a default authentication scheme, the SAML Provider is listed on the API Portal login page. Select the SAML Provider to open the SAML IdP login page. Provide the user credentials that are verified on the SAML IdP, and the user is logged in to CA API Developer Portal.

If the SAML Provider is set as default and you are unable to log in using SAML, use the *hostname/admin/login* URL to log in to API Portal and verify the SAML provider configuration.

NOTE

API Developer Portal does not support user creation and management in IdP. User management has to be done at the SAML IdP .

Having configured IdP with Portal, Portal administrators and Organization administrators can still create and manage users in Portal authenticated using CA APIM Authentication Scheme. For information about how to manage users from Portal, see the Get Started - User Types, Roles and Permissions section.

For solutions to troubleshoot issues that may occur while configuring the SAML authentication schemes, see the troubleshoot section. See our FAQ, sections for queries about the SAML SSO integration with API Portal.

For information about how to set up SSO for the API Gateway, see "Working with CA Single Sign-On" in the API Gateway documentation.

Establish Trust on SAML IdP

Collect the information that is required to establish trust from the Provider Configuration table. Ensure that the ACS URL provided is used to establish the trust.

The following values are required to establish trust on SAML IdP:

Information Type	Required Values
Service provider-specific information.	Requires the following values: • Assertion Consumer Service (ACS) URL URL where the SAML response is received from the IdP. • Service provider ID API Portal entity ID, or SAML request issuer. If the IdP does not have a service provider ID, use the default value that API Portal displays in the configuration screen.
API Portal-specific information:	Requires the following values: • SAML Token Attribute • SAML Token Attributeln

Edit SAML SSO Configuration

To edit the SAML SSO details:

1. Log in to the API Portal as an Administrator.
2. From the menu bar, select the gear icon, **Authentication**.
3. On the **Authentication Schemes** page, select the down arrow in the **Actions** section of a configured SAML SSO, and select **Edit**.
4. In the Edit Authentication Scheme page, select SAML SSO configuration to edit. For example, to edit the provider details, select the Provider Configuration option. Make the required changes and select **Save**.

   NOTE
   (Only for SaaS users) If the selected provider is **SAML SSO (new)**, the **Email Domains** field is read-only and you cannot edit it.

Delete SAML SSO Configuration

To delete the SAML SSO configuration:

1. Log in to the API Portal as an Administrator.
2. From the menu bar, select the gear icon, **Authentication**.

On the Authentication Schemes page, select the down arrow in the **Actions** section of a configured SAML SSO, and select **Delete**.

Troubleshooting

This section describes the solutions to troubleshoot issues that may occur while configuring the SAML authentication schemes.

Symptom:

Creating the SAML authentication scheme on API Portal throws the following error:

The specified username and password was invalid.

Reason:
The issue may be due to one of the following reasons:

- incorrect Identity Provider URL, or Issuer ID, or trusted certificate is provided as the provider configuration details.
- incorrect Assertion Consumer Service (ACS) URL, or Service provide ID is provided while establishing the trust on IdP.
- incorrect mapping of the Role or Organization attributes.

Solution:

Ensure the:

- provider configuration details are valid.
- service provider ID and ACS URL are similar to the one that exists on API Portal.
- role attribute that is mapped on API Portal is conceptually similar in your SAML IdP. The role attribute mapping that is returned in the SAML response should contain one of the roles that are mapped on API Portal as role attributes.
- organization that SAML response returns as part of organization attribute mapping must exist in API Portal.

If the issue persists after you have ensured all the values for creating authentication schemes are correct, we recommend re-creating the authentication scheme.

FAQ

This section contains Frequently Asked Questions and answers.

Frequently Asked Questions on Transitioning to Broadcom Okta Single Sign-On

- ***What is happening?***
  API Portal is transitioning its single-sign-on (SSO) capability to Broadcom Okta. Your users will continue to authenticate with your own identity provider (IdP), using the same SSO credentials they currently use. Your IdP(s) in API Portal application will need to be reconfigured to point to Broadcom's B2C Okta instance, and vice versa.
- ***Why?***
  Broadcom as a corporation has adopted Okta for SSO in order to make it easier for our customers to access all of Broadcom's SaaS products.
- ***How?***
  Layer7 Support/Product Management will work with your IdP administrator to transition your subscription to our new SSO solution. After your IdP and Okta have been configured to communicate with each other, you will have the opportunity to test the new SSO pathway with a small set of users before transitioning your entire subscription.
- ***Who?***
  Layer7 Support will work with your company's Portal Administrator(s), as well as your internal IT team who manages your SSO configurations.
- ***Is there any additional cost to migrate to use Broadcom SSO with OKTA?***
  No. Layer7 Support will work through this migration with customers at no charge.
- ***How does this affect our existing tenant in API Portal?***

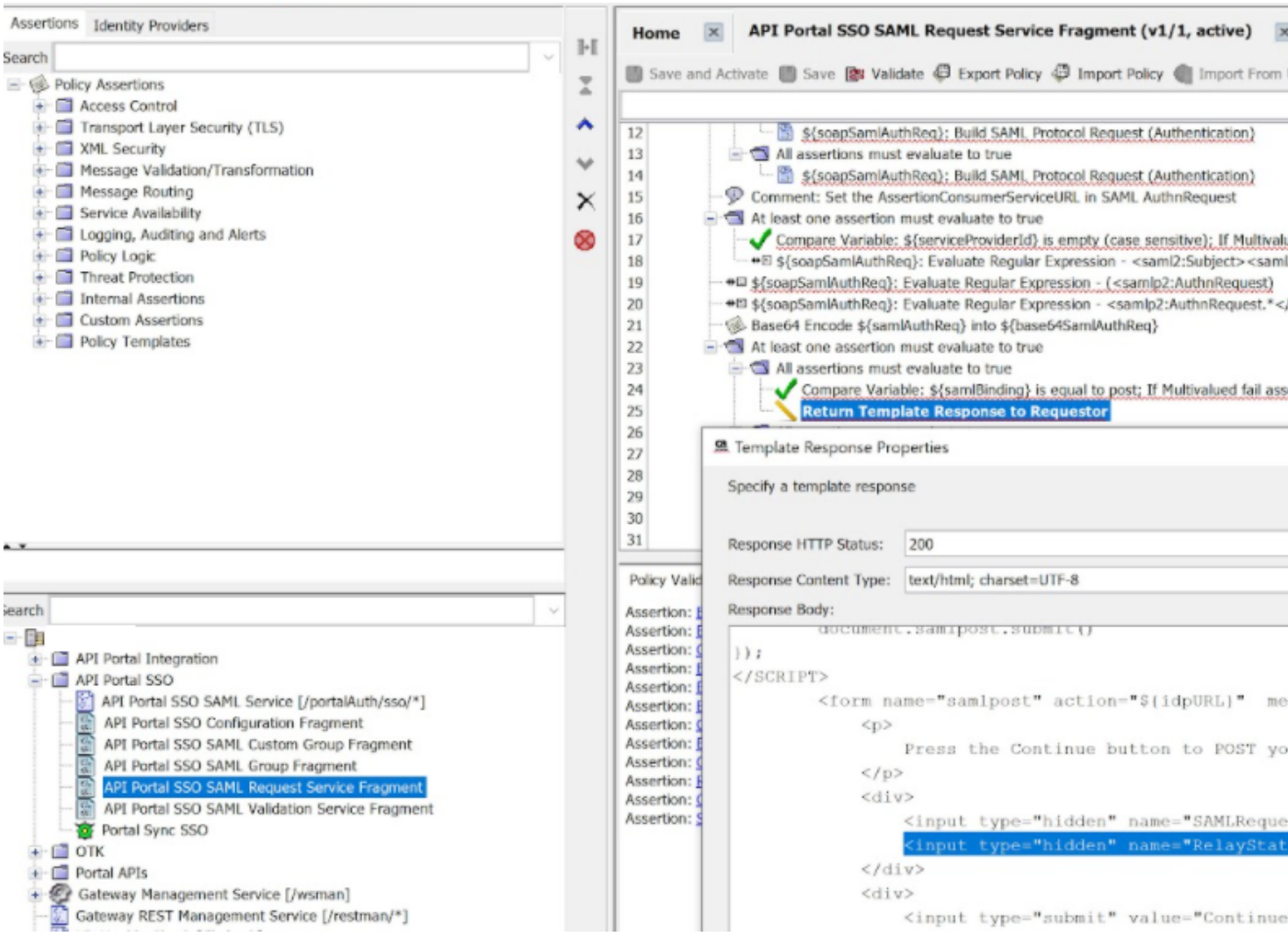
Once your IdP configuration details have been transitioned to Okta, there should be no change in how you access API Portal. You can continue to initiate a login either at your company's IdP or at API Portal. SSO users will continue to use their current credentials.

- ***Are there any benefits to this change?***
  Okta is a trusted, highly reliable SSO solution. If your company adopts other Broadcom SaaS applications, you should be able to use a single connection from your IdP to Broadcom's B2C Okta instance. This would eliminate the need to configure separate SSO connections for each application.
- ***Are there any detrimental effects of this change?***
  Your IdP administrator and API Portal administrator will need to work with Layer7 Support to perform the transition. Once that is complete, there should be no impact on your SSO users.
- ***Do we have to change our usernames and passwords?***
  No, your SSO usernames and passwords will continue to be stored by your own IdP, and do not need to change.
- ***What if we don't use SSO at all?***
  API Portal users that do not use SSO are not affected by this change. Such users would continue to log in directly to API Portal, no configuration changes are necessary, and this message may be disregarded.
- ***What if we use both SSO and API Portal native user authentication?***
  API Portal users that can authenticate by SAML SSO need to be transitioned as described above. When authenticating with SAML SSO, your IdP will then route users through Broadcom's B2C Okta. When authenticating directly with API Portal, there will be no change.
- ***If Okta is in the cloud, does this mean a malicious actor could access our tenant in API Portal?***
  All authentication redirection to Okta is encrypted with TLS security. Your users' SSO passwords are never transmitted to Okta. Only the Okta IdP configuration provisioned for your IdP will be able to authenticate the users of your own IdP. Administration of Broadcom's B2C Okta instance is strictly controlled by Broadcom IT.
- ***What do we need to do to make this change?***
  Contact Layer7 Support to initiate the process of transitioning to Broadcom Okta.
- ***Who do we contact if we need to make a change to our IDP configuration?***
  Contact Layer7 Support, who will interface with Broadcom IT if necessary.

SAML Frequently Asked Questions

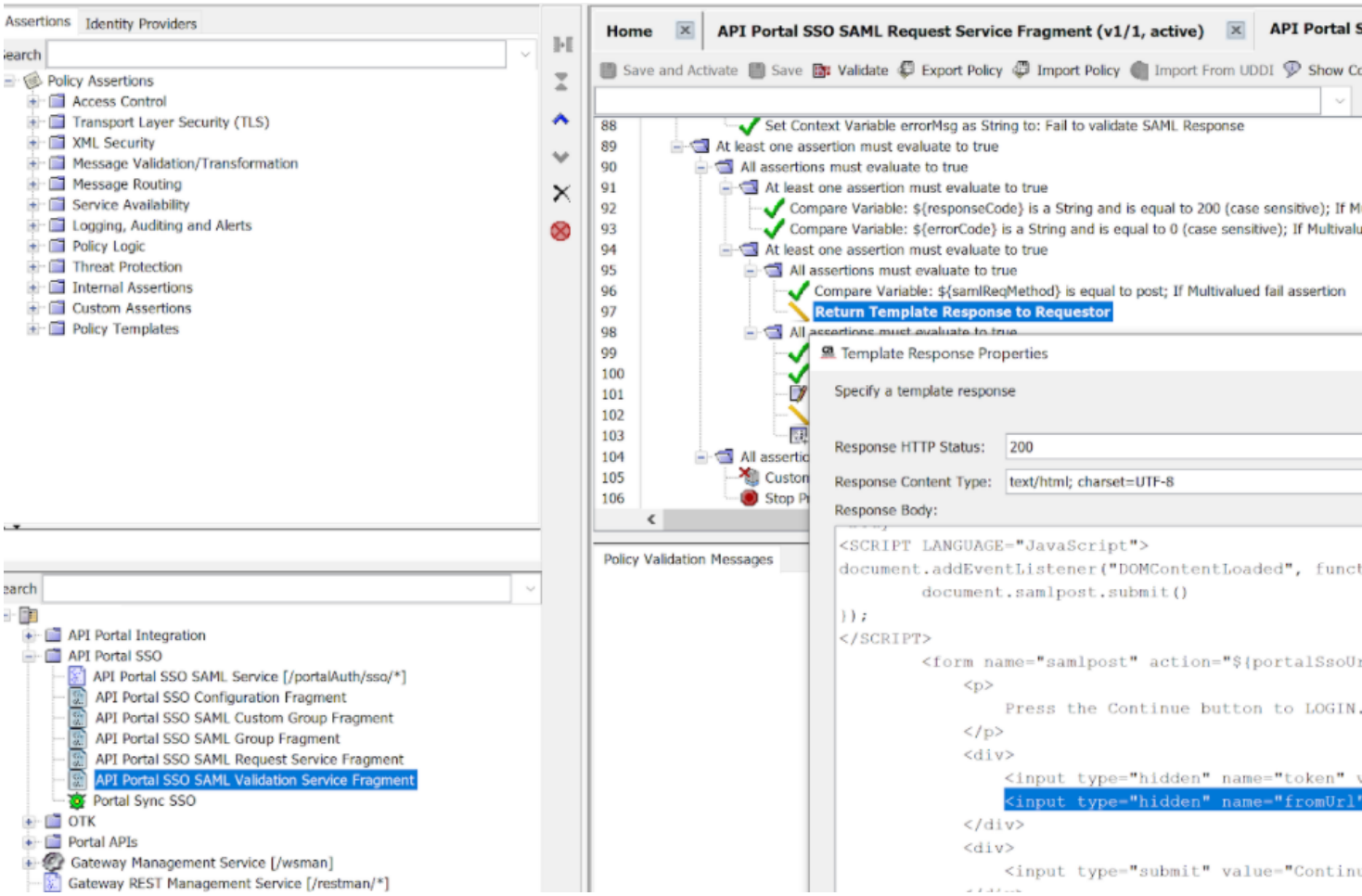
This section lists the Frequently Asked Questions regarding SAML integration with API Portal

- ***How can we go to the bookmarked page after SAML SSO authentication*?**
  When you access a bookmark page and log in into API Developer Portal, you will be redirected to the same bookmark page instead of the API Portal Dashboard or the admin page.
  If you are using the SAML SSO that has been deprecated leveraging the API Gateway and this functionality is not available after you upgrade API Developer Portal, ensure that the following changes are made in the **API Portal SSO** folder in API Gateway:
  – Edit the **Return Template Response to Requestor** assertion in **API Portal SSO SAML Request Service Fragment**, and add the following text in the **Response Body**:
  ```
  <input type="hidden" name="RelayState" value="${request.http.parameter.fromUrl}"/>
  ```



– Edit the **Return Template Response to Requestor** assertion in **API Portal SSO SAML Validation Service Fragment**, and add the following text in the **Response Body**:

```
<input type="hidden" name="fromUrl" value="${relayState}"/>
```



- ***Do we provide Just-in-Time Provisioning for SAML? How users are provisioned in API Portal from the IdP?***
  API Portal does not support user creation and management in IdP. User management has to be done at the SAML IdP. Assuming we are talking about Just-in-Time provisioning of user in API Portal, after being authenticated from the IdP, user attributes are passed on from IdP to API Portal with Authentication Response and based on attributes that are propagated, we provision user in API Portal.
- **I*s it possible to have different IdP per organizations? For example, organization 1 is federated with IdP 1, and organization 2 is federated with organization 2. Is it possible to have organizations with user accounts managed locally in the API Portal and others organizations federated with an IdP?***
  Portal supports multiple IdP configurations at tenant level, and not at organizational level.
- ***How are users created in the API Portal (Hybrid SaaS Portal) when using SAML Authentication?***
  When IdP is configured with API Portal, Portal administrators and Organization administrators can still create and manage users in Portal authenticated using CA APIM Authentication Scheme. For information about how to manage users from Portal, see the Get Started - User Types, Roles and Permissions section.
- **How are users managed, for example, when a user leaves their organization?**
  Users that are created in API Portal, the Portal administrator can delete the user profile. If there is IdP integration, the users are managed from the IdP side and the API Portal do not manage those users.
- ***What is the workflow for SAML assertion authentication?***
  After API Portal is integrated with SAML IdP, you can set the SAML authentication scheme as a default scheme. API Portal renders the selected SAML IdP login page to prompt for user credentials.
  If the SAML authentication scheme is not set as a default authentication scheme, the SAML provider is listed on the API Portal login page. Click the SAML provider to open the SAML IdP login page. Provide the user credentials that

are verified on the SAML IdP, and the user is logged in to CA API Developer Portal. The SAML response assertion is sent to API Portal and user is logs in to API Portal.

Set Default Authentication Scheme

Set an authentication scheme as a default scheme, and API Portal renders the default authentication scheme login page for user authentication. Setting a default authentication helps user log in without selecting an authentication schemes.

Follow these steps:

1. Log in as a Portal Admin.
2. From the menu bar, select the gear icon, **Authentication**.
3. On the **Authentication Schemes** page, select the down arrow in the **Actions** section of a configured authentication scheme.
4. Select the **Set as Default** option in the **Actions** section from the Authentication Schemes page.
   The authentication scheme is set as a default scheme.

Manage Password Policy

Password policy defines the rules for password creation. The policy is applied to a user account creation and during password change. API Management SaaS has default password policy. A Portal Admin can modify the password rules to enforce password complexity in API Management SaaS and can enforce users to employ strong passwords by defining password policies.

Follow these steps:

1. Log in as a Portal Admin.
2. From the menu bar, select the gear icon, **Authentication**.
3. On the **Authentication Schemes** page, for the CA APIM authentication scheme type, click the down arrow in the **Actions** section, and then select **Edit**.
4. In the Edit Authentication Scheme page, select the **Password Policy** option in the left navigation pane and configure the following rules:

Setting	Description
Minimum Password Length	Enter the minimum number of characters (8 through 60) required for the password.
Maximum Password Length	Enter the maximum number of characters (8 through 60) for the password.
Uppercase Characters	Set the number of uppercase letters that are required for the password.
Lowercase Characters	Set the number of lowercase letters that are required for the password.
Numeric Characters	Sets how many numbers (0-9) are required for the password.
Special Characters	Sets how many symbol characters are required for the password.

5. Define the Account Lockout Policy, and then click **Next**:
   - **Maximum Failed Attempts**
     Set the maximum number of allowed login attempts to safeguard against brute-force, or attempts to guess passwords. After the specified number of consecutive attempts, the user account is locked.
   - **Account Lock Duration**

Determines the number of minutes a locked-out user account remains locked out before automatically getting unlocked.

6. (Optional) To edit the CA APIM authentication details, select the **Basic Details** link on the left navigation pane. Specify the provider name, provider icon, Provider description, and then click **Next**.

   NOTE
   By default, CA icon is set as the provider icon. Provide a different PNG file to change the icon, and ensure that the file size must not exceed 500 KB.

7. Click **Save**.

Map IdP Users to Multiple Organizations

Use Case: You have created an authentication scheme, for example, LDAP. There are users (Publishers or Developers) who log in to API Portal using this authentication scheme. Now, you want to map some of the Developers to multiple organizations.

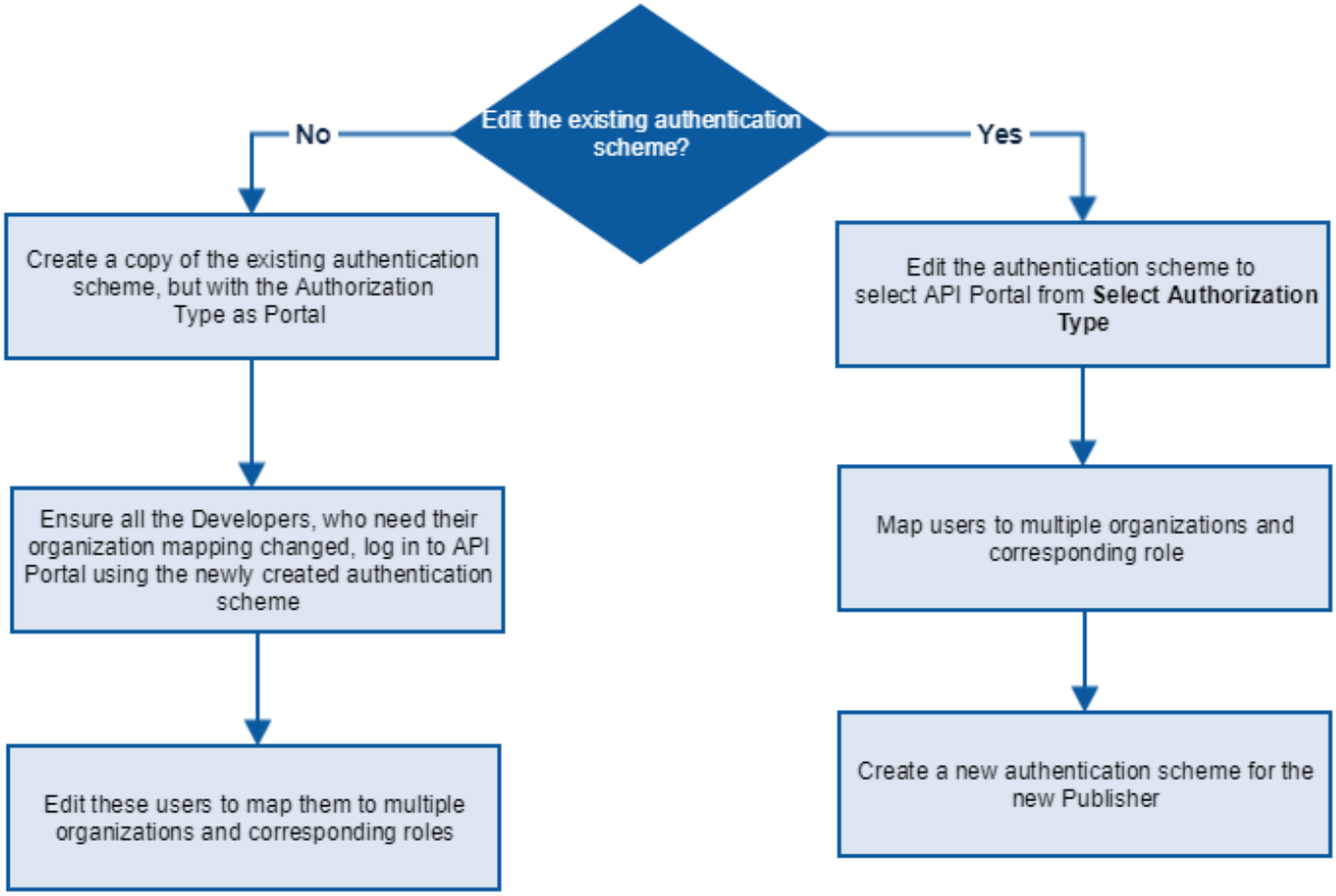
However, when a user logs in to API Portal using an external IdP, the organization and role is assigned according to the mapping provided by the administrator. This mapping is done during the configuration of the authentication scheme. This assignment cannot be altered as the values are derived from an external IdP.

Solution:

NOTE

- Other than organization and role, you cannot edit other details for an external IdP user.
- By default, all external authentication schemes have "Identity Provider" as the authorization type.

Figure 3: Map IdP Users to MultiOrg



User: API Portal Administrator

Exception: This feature is not available for the default authentication scheme and for "SAML SSO (old)".

Edit an Existing Authentication Scheme

You can manage the organization details of a Developer from API Portal by editing the authentication scheme and changing the authorization type to "Portal". Then, edit these users to map them to multiple organizations.

> **WARNING**
>
> After you change the authorization type to "Portal", all new *Publishers* of this authentication scheme are unable to log in to API Portal. To address this issue, create an authentication scheme for all the new Publishers. This issue is not applicable for users who have previously logged in to API Portal.

Follow these steps:

1. Edit the authentication scheme:
   a. From API Portal, select **Administration**, **Authentication**.
   b. Select **Edit** from the **Actions** menu of the authentication scheme.
   c. Go to **Attribute Mapping** section, and select **Portal** from **Select Authorization Type.**
   d. Save the authentication scheme.

2. Map users to multiple organizations:
   a. Select **Users.**
   b. Go to the **Developers** tab.
   c. In the **Actions** menu for the user, select **Edit**.
      The user details are displayed. This is a read-only page.
   d. Select **Next.**
   e. From the Select Organization and Role page, select the organization and the corresponding role.
   f. Select **Save** to save the mapping.
      The user is mapped to one or more organizations.
3. Create an authentication scheme for the new Publisher:
   a. Select **Administration**, **Authentication**.
   b. Select **Add Authentication Scheme.** Ensure the following field values are entered:
      a. Select the same authentication type, for example, "LDAP".
      b. Give a meaningful name for **Basic Details,** for example, "LDAP for New Publishers"
      c. Keep the **Select Authorization Type** as "**Identity Providers"**.
      d. Select a role from the available list. Map it to the following CA APIM Portal user roles that are similar to the user roles defined in your authentication scheme:
         - API Portal Administrator
         - API Owner
   c. Save the authentication scheme.
      The new Publisher must use this authentication scheme to log in to API Portal.

Create a Copy of the Authentication Scheme

You can manage users using the authentication scheme from API Portal by creating a copy of the authentication scheme with the authorization type set to "Portal". Then, edit the users mapping them to multiple organizations.

> **WARNING**
>
> After you create the authentication scheme, ensure all the Developers log in to API Portal in order to edit their organization mapping. A Publisher cannot use this authentication scheme to log in to API Portal. This method is best used when you have a limited number of Developers who need to have their organization mapping changed.

Follow these steps:

1. Create an authentication scheme:
   a. From API Portal, select **Administration**, **Authentication**.
   b. Select **Add Authentication Scheme.** Ensure the following field values are entered:
      a. Select the same authentication type, for example, "LDAP".
      b. Give a meaningful name for **Basic Details,** for example, "Portal-Managed LDAP".
      c. Change the **Select Authorization Type** to "**Portal"**.
   c. Save the authentication scheme.
2. Ensure all the Developers log in to API Portal in order to change their organization mapping.
3. Map users to multiple organizations:
   a. Select **Users.**
   b. Go to **Developers** tab.
   c. In the **Actions** menu for the user, select **Edit**.
      The user details are displayed. This is a read-only page.
   d. Select **Next.**
   e. From the Select Organization and Role page, select the organization and the corresponding role.

f. Select **Save** to save the mapping.
The user is mapped to one or more organizations.

Configure User Registration

Only registered users can access most API Portal features. This article describes how Portal Admins can configure user registration by:

- Specifying the Terms of Use that all API Portal users must accept when they register for an organization account.
- Enabling third-party registration, thereby allowing users to register themselves and their organization.

  NOTE
  Users can register themselves only if single sign-on (SSO) is disabled.
- (If third-party registration is enabled) Enabling the Registration Request Workflow for Third-party Registration, thereby requiring a Portal Admin to approve a third-party registration before the user can register for an organization account.

In this article:

Specify the Terms of Use

As part of the user registration process, new users must read and accept your enterprise's Terms of Use. The Terms of Use appear on the Account Setup page when users register for an organization account. Portal Admins can specify and edit the terms of use.

WARNING
The default Terms of Use are only an example. Ensure that you specify your enterprise's terms of use. You cannot leave them blank.

Follow these steps:

1. Log in to the API Portal as a Portal Admin.
2. From the menu bar, select the gear icon, **Registration**.
3. Replace the default terms of use with yours (using plain text), and then click **Save**.

Accept a User-Account Activation Invitation

1. When you receive a user-account activation invitation by email, open the message and select the activation link.
   Your web browser opens the Account Setup page.
2. Enter your information.

   NOTE
   If the username field is blank, enter a unique username. You cannot change usernames.
3. Read the **Terms of Use**, select the box next to **Accept Portal Terms of Use**, and then select **Activate Account**.
   API Portal opens and you are logged in.

Third-Party Registration

If the third-party registration is enabled, anonymous users can register themselves and their developer organization. Users cannot add themselves to an existing developer organization. If the Registration Request Workflow for Third-party Registration is also enabled, then a Portal Admin must approve the registration request. After getting approval, the user can finish registering for an organization account by completing the Account Setup page. Third-party registration is enabled by default.

If third-party registration is disabled:

- Portal Admins must register the users.
- Anonymous users cannot see the **Sign Up** button on the Home page or the **Sign Up** link on the navigation bar.
- Users who have registered for an organization account but have not yet activated their account remain in the User list.

> **NOTE**
> API Management SaaS includes security features that block bots attempting to register. One of those features is the **I'm not a robot** captcha on the third-party Registration page.

Enable and Disable Third-Party Registration

Follow these steps:

1. Log in to API Management SaaS as a Portal Admin.
2. From the menu bar, select the gear icon, **Registration**.
   The Edit Registration page appears.
3. Complete the following, and then select **Save**:
   - Select **Enabled** or **Disabled**.
   - To stop users from registering again using the same email address before activating the first registration, select **Limit User Registration** checkbox. For example, if a user uses xyz@abc.com to register, they will receive an activation email. If the activation is not done, they cannot attempt to register again with the same email address. This prevents spammers or robots from sending multiple activation emails and fraudulent actions.

Accept or Reject a Third-party Registration Request

Follow these steps:

1. Log in to the API Management SaaS as a Portal Admin.
2. From the menu bar, select the gear icon, **Requests**.
3. Select **View Details** next to the organization name.
   The Application Request page opens.
4. Review the information about the organization, and then select **Accept** or **Reject**.
5. If you select **Accept**, API Management SaaS sends an account activation email to the developer. If you select **Reject**, enter a message clearly explaining the reason, and then select **Send**. API Management SaaS sends the message to the developer.

Registration Request Workflow for Third-Party Registration

Anonymous users can use the following workflow:

1. The anonymous user completes the registration form.
2. If the Registration Request Workflow for Third-party Registration is disabled, the user:
   a. Receives the registration approval by email.
   b. Selects the link.
   c. Completes the account setup form.
3. If the Registration Request Workflow for Third-party Registration is enabled, the following occurs:
   a. The Portal Admin:
      a. Receives the registration request.
      b. Reviews and accepts or rejects it.

      > **TIP**
      > The user registration request indicates from which API Hub the user initiated the request.
   b. If the Portal Admin rejects the registration request, they can add a reason for the rejection in the registration rejection email. A registration approval or rejection email is sent to the user.
   c. The user receives the registration approval or rejection email.

d. If the user receives a registration approval email, the user clicks the link, and completes the account setup form. However, if the user receives a registration rejection email, the user can submit a new registration request, using the reason for the rejection in the email to improve the subsequent request.

Change the Domain Name in Registration and Activation Emails

By default, activation links in registration and account activation emails point to default internal domain or hostname. To ensure best security practices, we recommend that you change this default to a different domain.

Follow these steps:

1. In API Management SaaS, select **Portal API**.
2. From the **API** drop-down list, select **Portal API** (tenant name).
3. Select the **Settings** resource, then select **PUT**.
4. In the **input** field, enter your `CUSTOM_DOMAIN_NAME` .
5. Complete the **body** field. The following example uses `mycustomdomain.com` , Replace the value with your custom domain.

```
{
 "Uuid": "{{GENERATED_UUID}}",
 "Name": "CUSTOM_DOMAIN_NAME",
 "Value": "mycustomdomain.com"
}
```

NOTE
The UUID is auto-generated. If you need to see the UUID again, select GET for CUSTOM_DOMAIN_NAME.

6. Select **Submit**.
7. Ensure that the response code value is 200.

Manage Pending Accounts

When a user registers for a developer account, API Portal sends the user an account activation email. The account is then *pending*. On the Users page, the pending state icon

indicates pending accounts. To finish the self-registration process, the user selects the account activation link in the email, and then completes the Account Setup form. Users must complete this form before the account activation token expires.

Resend an Account Activation Email

While an account is *pending*, Portal Admins can resend the account activation email. This function helps when users cannot find, or mistakenly delete, the email. Pending accounts do not show the user name.

1. Log in to the API Portal as a Portal Admin.
2. From the menu bar, select the gear icon, **Users**.
3. On the **Actions** menu next to the user, select **Resend Activation Email**.

Delete a Pending Developer Account

Portal Admins can delete pending developer accounts. When you delete pending accounts, API Portal deletes the user records and revokes the account activation tokens.

On the Actions menu next to the user, select **Delete**.

Configure Request Workflow

As a Portal Admin, you can choose to enable or disable the following workflows:

- Registration Request Workflow
- Add Application Request Workflow
- Edit Application Request Workflow
- Delete Application Request Workflow
- Application Request Email Notification Workflow

Enable or Disable the Registration Request Workflow

Disabling this workflow allows new users to register without an approval process.

Follow these steps:

1. Log in to API Portal as a Portal Admin.
2. From the menu bar, select the gear icon, **Request Settings**.
3. Toggle the button for **Registration Request Workflow** to enable or disable, and then select **Save**.

Enable or Disable the Add or Edit Application Request Workflow

By default, the Add Application Request Workflow is enabled (approval by a Portal Admin is required for Org Admins or Developers to add applications) and the Edit Application Request Workflow is disabled (approval by a Portal Admin is not required for Org Admins or Developers to edit applications).

Disabling the Add Application Request Workflow allows Org Admins or Developers to create or edit applications without requiring a Portal Admin to approve or reject the application.

For more information about how to process application requests, see Application Requests.

Follow these steps:

1. Log in to API Portal as a Portal Admin.
2. From the menu bar, select the gear icon, **Request Settings**.
3. Toggle the button for **Add Application Request Workflow** or **Edit Application Request Workflow** to enable or disable, and then select **Save**.

Enable or Disable the Delete Application Request Workflow

By default, the Delete Application Request Workflow is disabled (approval by a Portal Admin is required for Org Admins or Developers to delete applications).

Enabling the Delete Application Request Workflow allows Org Admins or Developers to delete applications after acquiring approval from a Portal Admin. When the workflow is enabled, an email notification is sent when the attempt to delete an application is made. After application deletion workflow is approved, deletion of the application is logged in the audit log.

For more information about how to process application requests, see Application Requests.

Follow these steps:

1. Log in to API Portal as a Portal Admin.
2. From the menu bar, select the gear icon, **Request Settings**.
3. Toggle the button for **Delete Application Request Workflow** to enable or disable, and then select **Save**.

Enable or Disable the Application Request Email Notification Workflow

By default, the Application Request Email Notification Workflow is disabled. Enable this workflow to have API Portal send an email notification to the Org Admin when the Org Admin completes the Add Application form on the Add Application

page and when a Portal Admin approves the application. This option is available only when the **Add Application Request Workflow** is enabled. Publishers can review the notification and approve or reject the request.

Follow these steps:

1. Log in to API Portal as a Portal Admin.
2. From the menu bar, select the gear icon, **Request Settings**.
3. Toggle the button for **Application Request Email Notification Workflow** to enable the workflow, and then select **Save**.

Application Requests

Portal Admins can control which applications use specific APIs. If the Add Application Request Workflow setting is enabled (approval by a Portal Admin is required for Org Admins or Developers to add applications), then, when an Org Admin completes the Add Application form, a Portal Admin reviews and approves or rejects the application:

- If approved, the API proxy assigns a unique API key to the application and API Portal sends an email notification to the Org Admin.
- If rejected, the Portal Admin writes a message to explain the reason. Then the API Portal sends the message in an email notification to the Org Admin.

By default, approval by a Portal Admin is required for Org Admins or Developers to add applications but approval by a Portal Admin is not required for Org Admins or Developers to edit applications.

For more information about how to enable or disable the Add Application Request Workflow setting or the Edit Application Request Workflow setting, see Configure Request Workflow.

Process Application Requests

When an Org Admin completes the Add Application form on the Add Application page, if the Add Application Request Workflow setting is enabled, then the application is in the "Pending Approval" status. It remains in this state until a Portal Admin reviews and approves or rejects the add application request. Similarly, when an Org admin or Developer completes the Edit Application form on the Edit Application page, if the Edit Application Request Workflow setting is enabled, then the changes are not enabled until a Portal Admin reviews and approves the edit application request.

Portal Admins and API Owners do not require approval to add or edit applications.

Portal Admins can control which applications use specific APIs. If the Add Application Request Workflow setting is enabled, then, when an Org Admin completes the Add Application form, a Portal Admin reviews and approves or rejects the add application request:

- If approved, the API proxy assigns a unique API key to the application and the API Portal sends an email notification to the Org Admin or Developer.
- If rejected, the Portal Admin writes a message to explain the reason. Then the API Portal sends the message in an email notification to the Org Admin or Developer.

Follow these steps:

1. From the menu bar, select the gear icon, **Requests**.
   A list of requests display on the **Requests** page.
2. (Optional) Use the filter to restrict the list.
3. Select **View Details** next to the application request that you want to process.
   The **Application Request** page appears.
4. Do *one* of the following steps:
   – To accept the add/edit application request, select **Accept**.

API Portal sends a request approval message to the Org Admin or Developer who submitted the request. Also, the API proxy assigns a unique API key to the application (the default API key).

– To reject the add/edit application request, do the following steps:
  a. Select **Reject**.
  b. Enter a message that explains why you are rejecting the request, and then select **Send**.
  API Portal sends the message to the Org Admin or Developer who submitted the request.

The application request is processed.

Manage Requests from Developers

Requests can be registration requests or application requests:

- **Registration requests** are requests from Developers to register for an organization account. Portal Admins can manage registration requests by reviewing and accepting or rejecting them.
- **Application requests** are requests from Org Admins and Developers to get the API proxy to assign an API key to their application. Portal Admins and API Owners can manage application requests by reviewing and accepting or rejecting them.

In this article:

View a Request

You can view requests on the **Requests** page. This page displays the following information:

- The **Filters** option to limit the list of requests. You can filter by **Request Type** filter or **Organization Name** or both.
- The request type.
- The organization name.
- The originating source of the request, either API Portal or standard API Hub.
  Values:
  – **Portal:** The request originates from a Developer submitting a registration request for API Portal from API Portal.
  – **StandardAPIHub:** The request originates from a Developer submitting a registration request for API Portal from the standard API Hub.
- The date of the request.
- Options to accept or reject the request.

Follow these steps:

1. Log in to the API Portal as a Portal Admin.
2. From the menu bar, select the gear icon, **Requests**.
   A list of requests display on the **Requests** page appears.
3. (Optional) To view more information about a request, select **View Details**.
   The **Request Details** page opens.

   TIP
   To return to the **Requests** page, click the web browser's Back button.

Review and Accept or Reject a Pending Request

Follow these steps:

1. On the **Requests** page, in the **Actions** column, select **Accept** or **Reject**.

2. If you selected **Reject**, the **Request Rejected** dialog appears. Enter a reason for the rejection, and then select **Send**. The Developer can revise and resubmit the request, using the reason for the rejection in the email to improve the subsequent request. Otherwise, selecting **Accept** accepts the request.

Registration Request Workflow

If a Portal Admin accepts a registration request, API Portal sends the Developer who sent the registration request a standard email message containing a link to register for an organization account. The Developer clicks the link, and then completes the account setup process. The **Account Setup** page opens in API Portal or API Hub depending on where the registration request originated.

If a Portal Admin rejects a registration request, the **Request Rejected** dialog displays. The Portal Admin can send an email message to the Developer from this dialog. Add an explanation for the rejected request in the email that clearly states the reason for the rejection. When the Developer receives the email, they can revise and resubmit the registration request. In resubmitted registration requests, the **Request Details** page shows the reason for the rejection in the email that API Portal sends to the Developer.

Application Request Workflow

When the Portal Admin accepts an application request, the Developer who sent the registration request receives an API key for the application. The Developer must add the key to their application so that the application can consume your APIs.

Configure Mail Server at Tenant Level

You can send emails from a different mail server with custom SMTP configurations. This is done at the tenant level through the *Email/SMTP Settings* option. You can use trusted certificates to authenticate the API Portal or client, or both.

> **NOTE**
> Ensure that the tenant uses only the following SMTP authentication mechanisms, as supported by API Portal:
>
> - PLAIN
> - LOGIN
> - CRAM-MD5
> - DIGEST-MD5

To configure SMTP at the tenant level:

1. Log in as administrator.
2. From the menu bar, select the gear icon, **Email/SMTP Settings**.
3. Configure the following options: **SMTP Configuration**, **Connection Details**, and **Email Options**.

SMTP Configuration

Option	Action	Notes
Custom SMTP Service	Select **Enabled**.	Disabled option is used for the default mail server that is configured during deployment.
Protocol	Select from **SMTP**, **SMTPS**, or **SMTP TLS**.	

If SMTPS or SMTP TLS is selected:

Option	Action	Notes
SSL Authentication Type	Select **Server Authentication** or **Mutual Authentication**.	• Select Server Authentication if you want API Portal to send the client a trusted certificate to authenticate itself. • Select Mutual Authentication if you want the client and API Portal to mutually authenticate each other using their corresponding trusted certificates.
Server Certificate	Click **Choose File** and upload a trusted certificate in X.509 format that is required for a secure connection with the SMTP server.	
(For Mutual Authentication only) **Client Certificate**	• Click **Create CSR** and fill in these values, then hit **Create**: – Common Name – Alias Name – Organization/Department/City/State/Country – Key Size • Choose an **Alias Name** from the dropdown and upload a trusted certificate.	Notes: • Common Name specifies a distinguished name that is associated with your CSR. Recommendation: The Common Name is typically composed of Host + Domain Name. • Alias Name specifies a common identifier name that is associated with the CSR. Ensure that you add a unique alias name every time you upload a new certificate. • Organization/Department/City/State/Country specifies the details relevant to your organization. • Key Size: Select the key length (in bits) for the RSA Key pair. The signature algorithm that is used to generate the key pair is SHA256withRSA.

NOTE
Regarding certificates:

- The maximum file size of the certificate must be 50 KB.
- Ensure that the format of the certificate and file type is valid. If you upload an invalid certificate, selecting **Save** does not save the file.
- If you have already uploaded a certificate to authenticate your client, then the newly uploaded valid certificate replaces the old one after you select **Save**. This change cannot be reversed.
- Replacing a previously uploaded certificate may disrupt your existing SMTP connection. In such a case, API Portal displays a warning message. Ensure that you check the corresponding CSR, the connection details, and so on, to establish a successful SMTP connection.
- If you have already uploaded a client certificate, you cannot delete it. You need upload a new one.
- Do not upload an expired certificate.
- CSR is not available for download later.
- If you have previously configured a connection, API Portal continues to connect to that connection.
- If you do not have a previously successful connection, the connection to the SMTP server shows as inactive.
- To delete the saved certificate, select **Clear File**, and then save the changes. This is applicable only for server certificates.

Connection Details

Define the connection details for an SMTP server.

Option	Description
SMTP Host	Specifies the Host Name of the SMTP Server.
SMTP Port	Specifies the port of the SMTP server through which the communication happens. Layer7 recommends that one of the following common SMTP ports be used: • SMTP (587) • SMTPS (465) • SMTP TLS (587)
(Optional) **Username**	Specify the user name if the SMTP server is enabled for authentication.
(Optional) **Password**	Specify password that is associated with the user name.

Email Options

Define the emails options. Note that the domain associated with the Sender's Address and Bounce Email must be a trusted domain on the SMTP host.

Option	Description
(Optional) **Sender's Name**	Specifies the name of the sender.
Sender's Address	Specifies the from email address.
Verification Email	Specifies the email to test if the connection is successful.
(Optional) **Bounce Email**	Bounced email notifications are sent to the specified email address.

Select **Save** to save your configurations.

Verify that configuration is successful, as follows:

- After saving, API Portal tests the connection to the SMTP server.
- If the connection is successful:
  - A success message stating "Connection is active" is displayed.
  - A test email is sent to your specified verification email address.
- If the connection is unsuccessful:
  - API Portal allows you to save the configuration but there is no connection to the SMTP server.
  - A warning message stating "Connection is inactive" is displayed.

NOTE
API Portal's connection to the SMTP server is validated each time an email is sent. If the email is received successfully, it means the connection is successful and accordingly the connection status will display on the Email/SMTP Settings page. Similarly, even if the connection had been up and running but a delivery error occurred, the connection is now found inactive. This status is updated on the Email/SMTP Settings page.

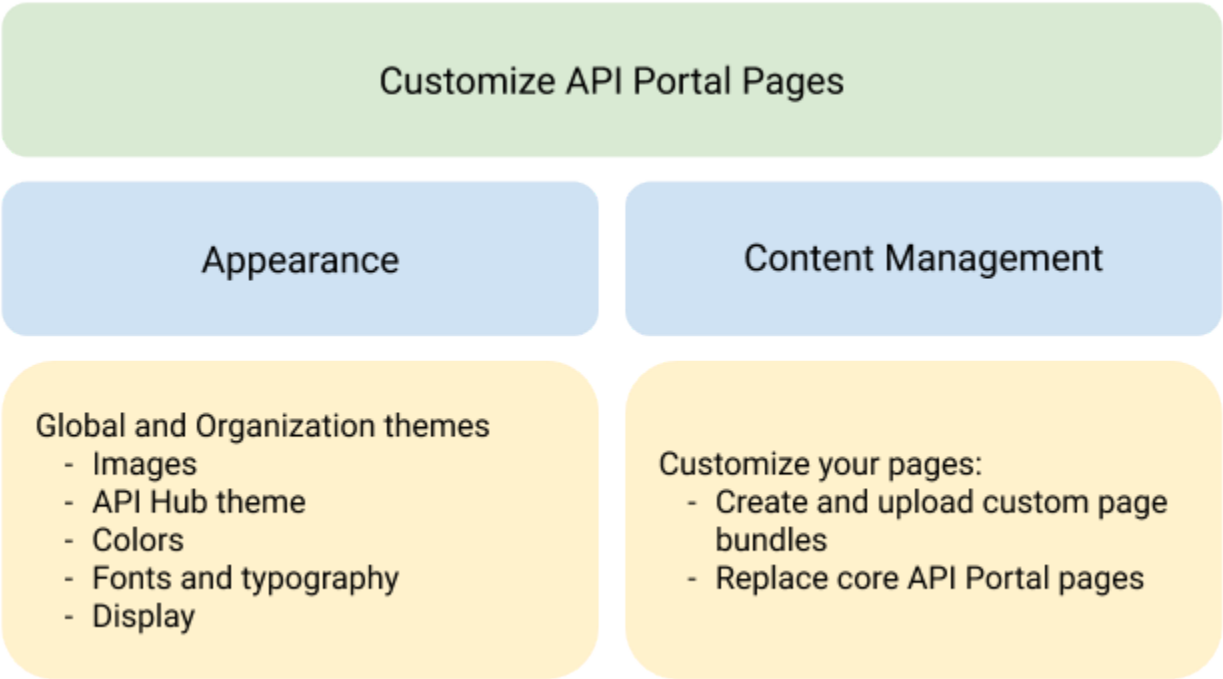
Customize API Portal Pages

ATTENTION

Custom Pages for Portal Versions 5.1.2 or Newer: Inline JavaScript Prohibited

Content-Security-Policy is a HTTP response header that web browsers use to enhance the security of a document or web page. To comply with current updates in the Content-Security-Policy header, inline JavaScript is now blocked from being executed in a web browser. If you have any existing custom pages or plan to have custom pages that contain inline JavaScript, you must extract those JavaScript elements into a separate script and reference it as an external source.

A Portal Admin can perform API Portal customization. To understand API Portal customization options, see the following diagram:



Change the Page Appearance

Customize the look and feel of your API Management console. From the menu bar, select gear icon, **Appearance**. For more information, see Customize Page Appearance.

Customize Page Appearance

You can customize the look of your API Management SaaS pages and API Hub by modifying the appearance components.

Apply your customization to all organizations by modifying the global theme. Apply customization to a specific organization by selecting the organization.

Customize the Page Appearance

Follow these steps:

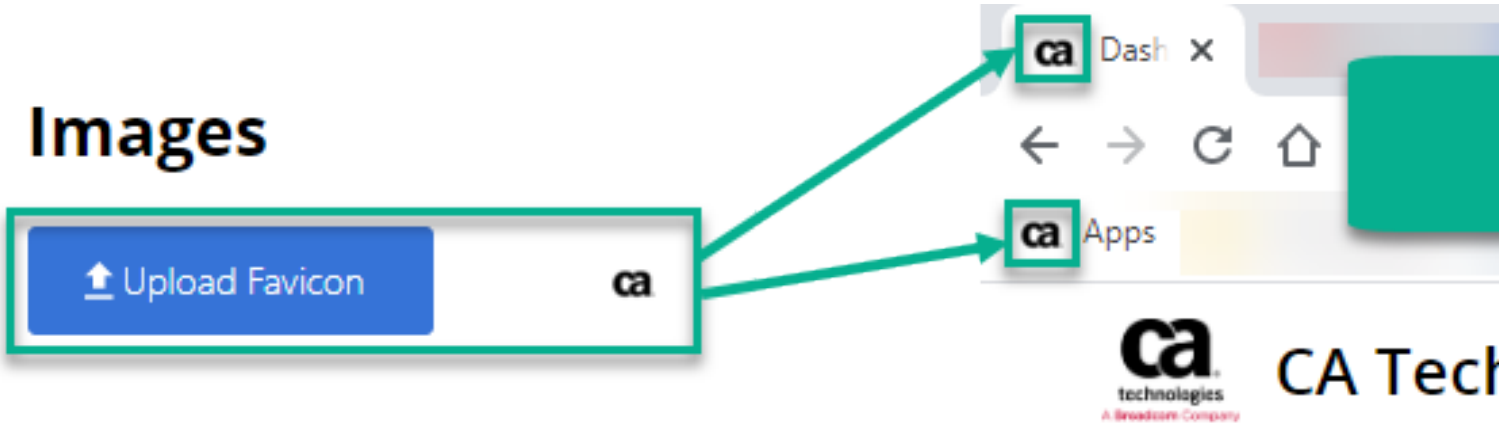
1. From the menu bar, select the gear icon, **Appearance**.
2. To customize the page appearance for:
   – All organizations, select **Manage Global Theme**.
   – A specific organization, select the organization.

The Appearance page appears.

You can modify the following appearance components:

Images

In the **Images** section of the Appearance page, you can select and upload the favicon for your API Management SaaS pages, as shown in the following example. To avoid a white box around your logo, make the PNG background transparent.



API Hub

In the **API Hub** section of the Appearance page, you can select colors and typography for the standard API Hub.

> **NOTE**
> The following settings are shared between API Management SaaS and API Hub:
>
> - Logo and favicon
> - Background color
> - Link color
> - Link Hover color

For more information about API Hub, see API Hub.

Colors, Font Sizes, and Typography

In the **Colors**, **Font Sizes**, and **Typography** sections of the Appearance page, you can select colors, font sizes, and typography for API Management SaaS.

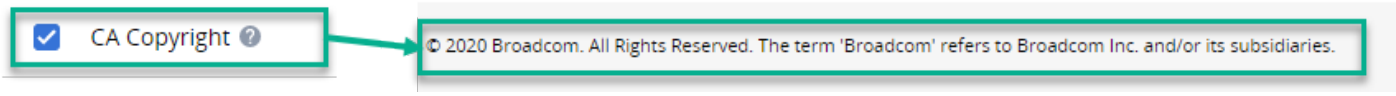
> **NOTE**
> The following settings are shared between API Management SaaS and API Hub:
>
> - Logo and favicon
> - Background color
> - Link color
> - Link Hover color

Display

In the **Display** section of the Appearance page, you can define what is displayed on the login screen and in the page footer in API Management SaaS.

You can do the following:

- Display the API Management SaaS version number on the login screen by selecting the **Version Number** checkbox.
  Default: Selected
- Display the CA copyright information on the page footer by selecting the **CA Copyright** checkbox.
  Default: Selected



- Show a banner displaying the cookie notification in API Management SaaS along with the link to the privacy policy by selecting the **Cookie Consent** checkbox.
  Default: Cleared
  Follow these steps:
  a. Select the **Cookie Consent** checkbox, and then select **Edit**.
  b. Complete the following, and then select **Apply Changes**:
     a. Select the font and font size.
     b. (Optional) Replace the default text or add your own.
        Maximum word limit: 400 characters
     c. Set hyperlinks (if any).
        Maximum hyperlinks: Two
     d. Select the cookie position (Bottom or Top) and theme (Dark or Light).
     e. Click **Preview** to preview your cookie consent.
     This banner is displayed in all in API Management SaaS pages until the user hides the banner (selects the close icon).

Restore the Theme to the Defaults

On the Appearance page, scroll to the bottom of the page, and then click **Restore Defaults**. All values are reset to the out-of-box defaults.

Customize Core Pages

This article describes how to customize the core pages of API Portal.

What You Can Customize

You can fully customize the following core API Portal pages to provide your customers with a fully branded experience of the public page:

- Home
- Login
- Sign Up
- Account Setup
- Reset Password
- New Password

You can bundle your custom pages (and all the related files, for example, js and css) in a zip file. Upload this file to API Portal and apply the customization, instantly.

The **Content Management** menu provides an easy way to set your custom pages as the default. Your current home page is automatically listed as "default".

You can either use the templates provided in the Content Management page or use your own html content to apply the customization.

Use Templates for Customization

API Portal provides a set of templates for each of the core API Portal pages that you want to customize. Each template file contains the minimum content that is required to make the corresponding page work as expected. You can create your own custom pages by customizing the template files or by creating your own files from scratch. The templates are available in `SampleTemplates.zip` file that you have to download from **Content Management, Upload Page Bundles** section.

The zip file contains six html pages and folders containing corresponding js and css files. This zip also contains a common folder where you can put common resources like js and css files, which can be applied across all pages.

Templates ensure easier and faster customization.

> **WARNING**
> The template file works similar to the default API Portal page that you want to customize. Ensure that you understand the functionality before attempting to change the files. Random changes in the code might result in an unstable page.

Prerequisites for Customization

Verify the following prerequisites before customizing the core pages:

- You have Tenant or Portal Administrator access in API Portal.
- You are aware of the external authentication schemes that must be applied on the Login page.
- You know that the permissible size limit for the zip file should not exceed 15 MB.
- The following file types are supported:

  - png
  - jpg
  - jpeg
  - gif
  - ico
  - svg
  - html
  - css
  - js
  - ttf
  - woff
  - woff2
  - otf
- File names (including the zip file) should only contain alphanumeric characters, underscores, spaces, and hyphens.
- You have experience in coding and consuming the APIs. Read the APIs that are applicable for the page you want to customize. The following table lists the minimum APIs that you can consume to make the pages work as expected:

Page	APIs	Description
Login	`/api/{tenantId}/authenticate/login`	To authenticate users using API Portal, call the POST method on this API with payload containing user credentials.
	`/admin/public/auth/schemes`	Provides a list of authentication schemes available for the tenant. These authentication schemes are displayed in the Login page.
	`"/api/{tenantId}/authenticate/getPublicKey`	'If the API returns the public key, then the password has to be encrypted in the "/api/{tenantId}/authenticate/login" payload.'
Sign Up	`/admin/Portal.svc/Registrations`	To place a sign-up request, call POST method on this API with payload containing user information.
Account Setup	`/admin/accountSetup`	Account Setup page is loaded with a token as a query parameter, which is required for setting up the account. To validate the token, call GET method on this API with token as query parameter.
	`/api/{tenantId}/authenticate/getPublicKey`	If the API returns the public key, then the password has to be encrypted in the "/admin/accountSetup" payload.
	`/admin/accountSetup`	To complete the account setup process, call PUT method on this API with payload containing user information, password, and token.

Reset Password	`/admin/Portal.svc/ResetMyPassword()`	To place a request to update the password, call GET method on this API with username as query parameter.
New Password	`/admin/passwordResetTokenValidate`	This page is loaded with a token as path parameter, which is required in the password reset process. The token can be validated by calling GET method on this API with token as query parameter.
	`/api/{tenantId}/authenticate/getPublicKey`	If the API returns the public key, then encrypt the password in the "/admin/UpdateMyPassword" payload.
	`/admin/UpdateMyPassword`	To complete the reset password process, call POST method on this API with payload containing password and token.

APIs to Create Custom Pages

Ensure you are aware of the APIs required to make the custom pages work, as expected.

For example, to customize the Login page, you need text boxes for entering the username and password, display the authorization schemes (portal, LDAP, SAML, and so on), and a **Login** button. The user must select an authorization scheme, and then enter the credentials. When the user clicks **Login**, API Portal calls an API to validate the user credentials.

> **NOTE**
> The following Swagger representation is created from a static JSON file.

Prerequisites for Re-customization

You do not need to remove an existing bundle to upload a new one. System allows you to upload another bundle, while the existing one is still in use. Consider the following points for re-customization:

- You cannot upload more than two bundles. To add one more bundle, delete the bundle that is not in use.
- **Recommendation:** Use a different html filename every time you apply the customization on the same page.
  For example, if you have used home.html for the initial customization, then use home1.html for the subsequent customization. This process ensures that the page caching is considerably improved and the changes are reflected faster.
- **Recommendation:** Keep a backup of your previous customization, if necessary.

Start Customizing

1. Click the gear icon on the upper right corner of any Portal screen to access the main menu and select **Custom Pages**.
2. To use the templates, download and unzip "SampleTemplates.zip" from **Upload Page Bundles**.
   a. Go to **<PageName>**.
      The folder contains the html, js, and css files.
   b. Depending on your organization's needs, customize the files. Read the APIs that are applicable for the page that you want to customize.
   c. Save your changes.
3. Create a zip file with all the contents that you need for customization.
4. From **Upload Page Bundles**, select **Upload Select Upload ZIP Bundle** to upload this zip file.

   > **NOTE**
   > If you have already uploaded two bundles, you have to delete an unpublished bundle to upload a new one.

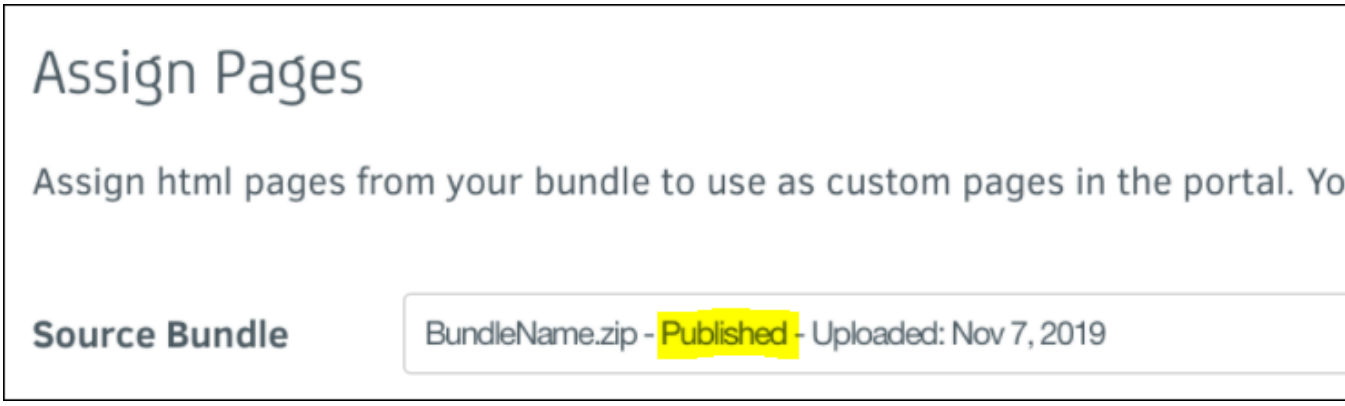
5. Select the html file from the **<PageName>** drop down. For example, if you are customizing the Home page, select the Home drop-down.

   NOTE
   All the assignments should be from a single bundle; pages from different bundles cannot be saved.

6. Select **Save & Publish**. Selecting **Cancel** restores the previous customization, but the recently uploaded bundle remains in the Source Bundle drop down.

   NOTE
   Consider the following points after you have customized:

   - The bundle status changes to "Published" as shown in the following screenshot.



   - You have selected a bundle, but assigned and saved all the pages to portal default. The selected bundle becomes unpublished.
   - The Delete Bundle link does not appear when you select a published bundle from the Source Bundle drop down.

Reset to Default Settings

Select **None** from Source Bundle drop down and save your changes. All the pages are reset to default portal pages and the published bundle becomes unpublished.

Delete a Bundle

You can only delete a bundle that is unpublished (not in use).

Follow these steps:

1. Select the bundle that is not in use from **Source Bundle**.
2. Select **Delete Bundle**.
3. Select **Delete**.

FAQs

- I have customized a page. How do I revert to the default page?
  You can select the "default" option from the drop-down.
- To re-customize the pages, I uploaded a new bundle. I selected the pages from the drop-down. But I did not save the changes and navigated to another page. Will the older customization be retained?
  Yes. Customization does not change until you publish or save the changes. Similar applies when the Content Management page times out before you save the changes.
- I customized a page using a file, but the changes are not reflecting immediately or are taking longer than expected. What do I do?

This problem occurs because of caching. You may have used the same filename when reapplying the customization. The resources are updated, but the dispatcher still serves the old resources. It might take an hour to serve the updated content due to caching on dispatcher.

To force the dispatcher to serve the updated content instantly, the Portal administrator can restart the dispatcher container.

> **WARNING**
> API Portal is unavailable while the dispatcher restarts.

Now, when the dispatcher starts serving updated pages, you still might not see the updated content. This is because of web browser caching and might take a day to show the newer version. Clear the cached assets for the API Portal site or open the site in private or incognito mode.

- I applied the changes to the Login page and I can see the customizations. But something went wrong in this page, because I am unable to relogin and fix the changes. How do I go back to the default Login page?

  This issue might have happened because you modified the `main.js` file incorrectly. Ensure that you read the APIs before attempting to change the code. To go back to default Login page, enter http://<portal.ca.com>/admin/login into your web browser.

Manage Custom Fields

Custom fields provide API and application metadata that API Gateway administrators can use in their policies by way of context variables. The metadata can give policies more flexibility for processing messages between applications and APIs.

For example, a policy can use the value of an API custom field. The policy can use the value to route requests to the API on either the production server or the sandbox server.

As another example, a policy can use the value of an application custom field to return:

- A product coupon if the request came from a mobile app.
- A product catalog if the request came from a desktop application.

For more information about how to write policies and how to use context variables, see "Context Variables" in the API Gateway online documentation.

In this article:

Example: Add and Use a Custom API Field

The following example illustrates how Gateway administrators and API publishers can work together to add and use a custom API field:

1. During planning, API owners and Gateway administrators decide they want the Gateway to cache calls that are sent to select APIs to prevent redundant calls from overwhelming those APIs. They decide to add a custom *API* field to the API Portal. For each API, the API owner uses the custom API field to specify whether to cache calls to the API. A context variable in the policy returns the value of the custom *API* field by way of a standard service property.
2. During implementation:
   a. The API owner adds the `API Cache` custom API field, and then gives the field that the value set Yes/No.
   b. The Gateway administrator adds the `${service.property.APICache}` context variable to a policy, and then writes policy that caches calls to the API if the value of the `${service.property.APICache}` context variable is *Yes*. Every 30 seconds, the policy retrieves calls from the API cache and sends them to the API.

      > **TIP**
      > Administrators can review the names and values of custom *API* fields that are assigned to APIs. For more information, see the "View Metadata Assigned to APIs" section.

3. The API owner publishes the `Billing` API on API Portal and uses the `API Cache` custom API field to assign the value *Yes* to the API.
4. The next time that the Gateway synchronizes with API Portal, the Gateway gets the value that is assigned to the `Billing` API (*Yes*) and assigns it to the `${service.property.APICache}` context variable.
5. During run time, when the Gateway receives calls to the `Billing` API, the policy sends the calls to the `Billing` API cache.

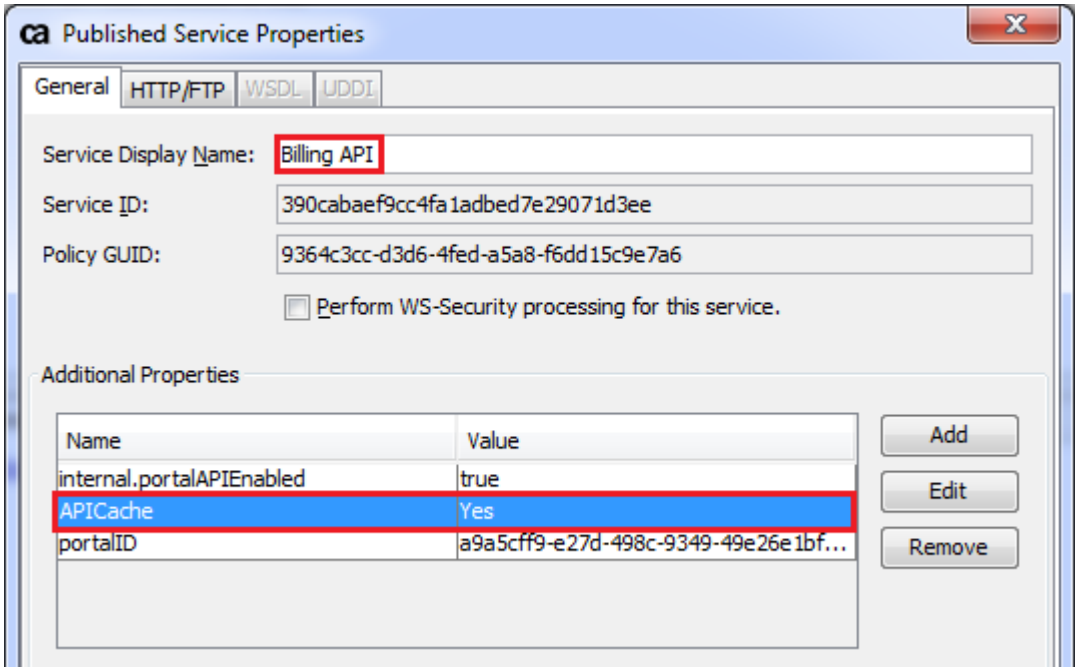
View Metadata Assigned to APIs

You can view metadata assigned to an API using the API Gateway and Policy Manager.

> **NOTE**
> You can also list the names and values of metadata assigned to an API using the API Portal or by way of API Portal.
>
> For more information about the Portal API (PAPI), see Portal API (PAPI).

Follow these steps:

1. Log in to the API Gateway as an administrator.
2. Locate the API in the **Portal APIs** folder.
3. Right-click the name of the API, and then select **Service Properties**.
   In the Published Service Properties dialog, the General tab lists the custom API fields and the assigned values for the API. The following image shows an example of the General tab:



Example: Add and Use a Custom Application Field

The following example illustrates how Gateway administrators and API publishers can add and use a custom application field:

1. During planning:
   a. API Owners and Gateway administrators decide to support the ability of their APIs to return different kinds of content to web applications based on whether the applications calling the API are built on mobile or desktop browsers.

   b. Hence, they decide to add a custom application field to the API Portal. This custom field requires that the Org Admin or Developer to specify the browser type for each of their applications.
2. During implementation:
   a. The API Owner adds the `Browser Type` custom application field and gives it the value set *Mobile/Desktop*.
   b. The Gateway administrator writes policy that, when a mobile app calls the `Product` API, requests product coupon data from the product database and returns it to the mobile app. If a desktop application calls the `Product` API, then the policy requests product catalog data from a database and returns it to the desktop application. The administrator uses the `customFieldsMetadata` context variable to integrate the custom field with the policy. For more information, see the "Integrate Custom Application Fields with Policies" section.
3. An Org Admin adds the `Dog Food` mobile application to API Portal and uses the `Browser Type` custom application field to assign the value *Mobile* to the API.
4. The next time that the Gateway syncs with the API Portal, the `Dog Food` API key carries the custom field name and value to the Gateway, which saves it in the OTK database.
5. When the Gateway receives calls from the `Dog Food` mobile application to the `Product` API, the policy gets the name and value of the application `Browser Type` custom field in the `customFieldsMetadata` context variable from the OTK database. Because the value of the application `Browser Type` custom application field is *Mobile*, the policy requests product coupon data from the product database and returns it to the app.

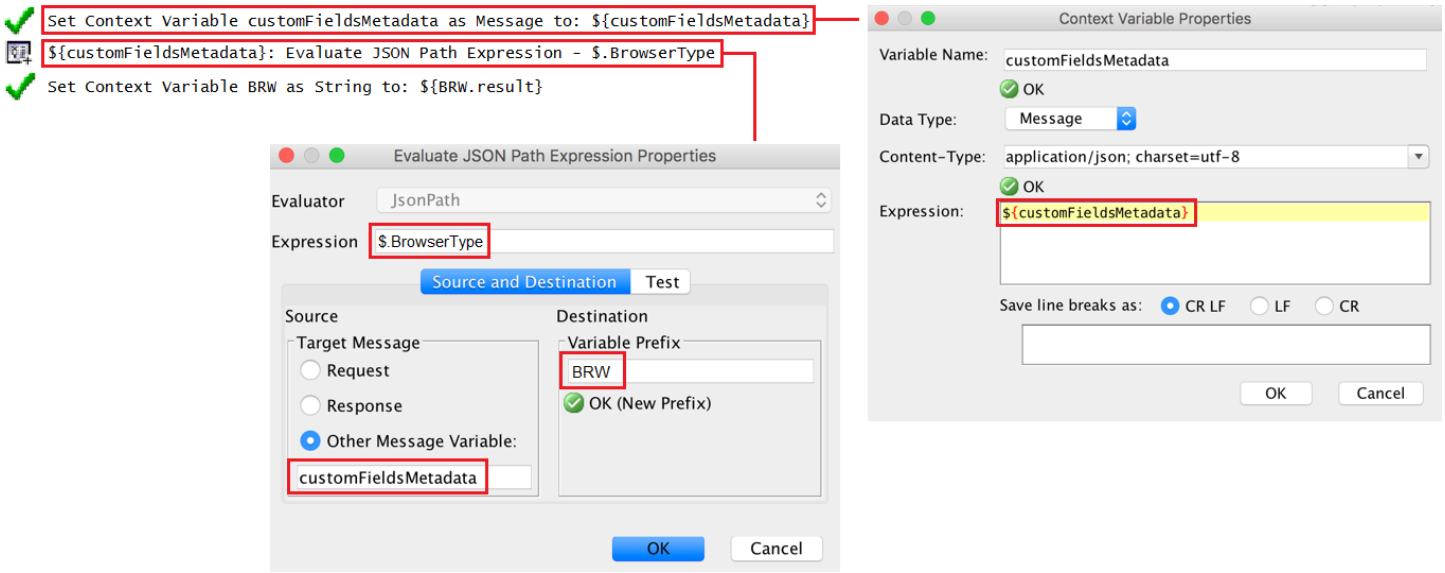
Integrate Custom Application Fields with Policies

Include the following steps in the policy:

1. Retrieve the `customFieldsMetadata` context variable. This context variable contains the name and value in JSON format for all custom application fields in an application.
2. Evaluate the `customFieldsMetadata` JSON value to extract a specific custom application field.

   NOTE
   The names of custom fields are concatenated.
3. Assign the value of the custom field to a context variable.

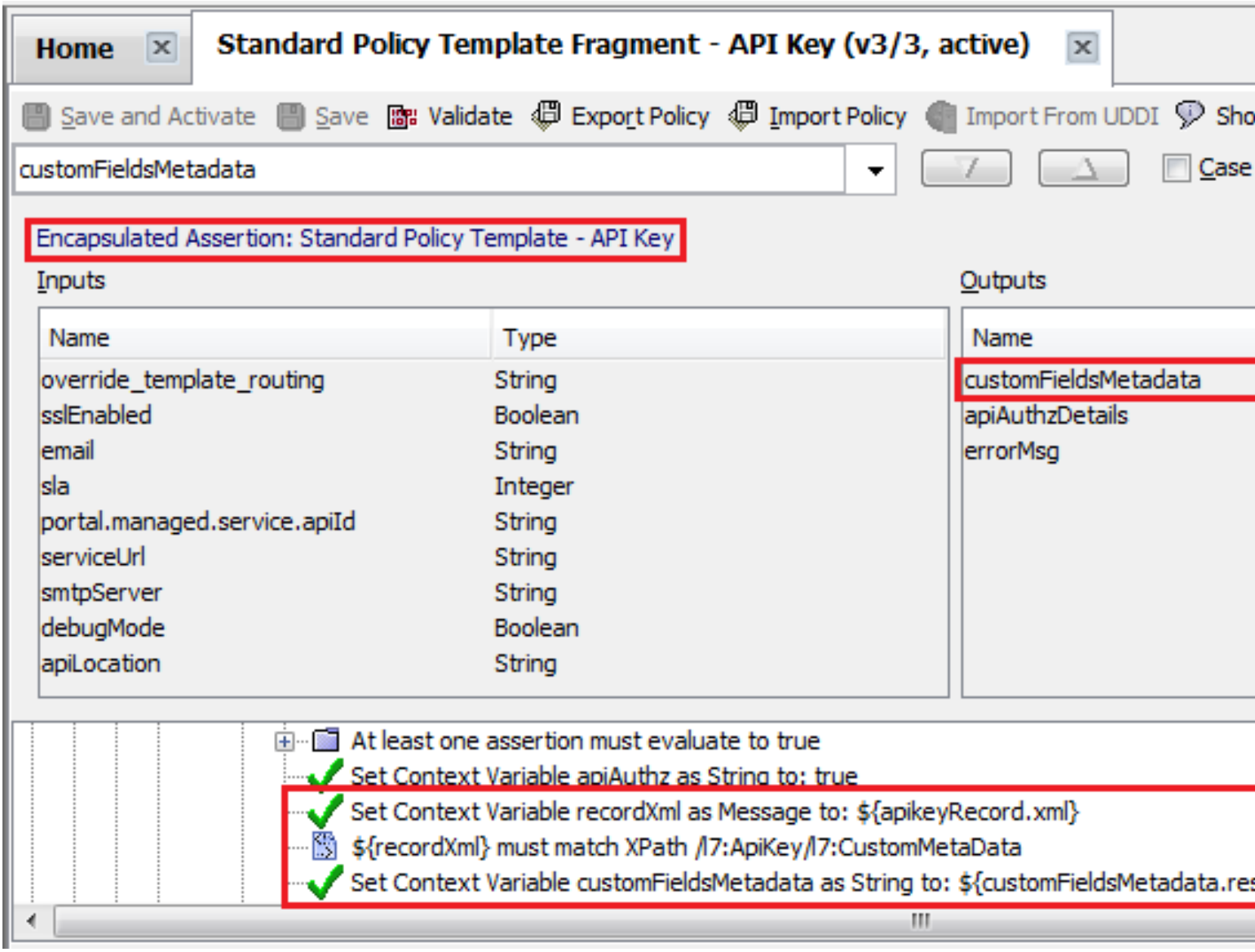
The following figure shows an example policy snippet that integrates the `Browser Type` custom application field into a policy:



To view the metadata that is assigned to an application, use the API Portal or the Portal API.

Custom Application Fields and Policy Templates

The API Portal standard policy templates set the `customFieldsMetadata` context variable. The following image shows the relevant snippet for the `Standard Policy Template Fragment - API Key`. If API Gateway administrators customize, or create, policy templates, they must ensure that their policy templates set this context variable.



Manage Custom Fields

You can manage add, edit, enable, disable, and delete custom fields in the API Portal. Portal Admins and API owners can add custom API and application fields to the API Portal. Custom fields provide a menu of options or a text input field. After an administrator or an API owner adds a custom *API* field the API Portal, when API developers add or edit an API, they can assign a value to the API using that custom field. After an administrator or an API owner adds the custom *application* field the API Portal, when API developers add or edit an application, they can assign a value to the application using that custom field.

WARNING
Custom *API* fields are available only to APIs that are published on the API Portal. If the API is published on the API Gateway, the fields are not available to the API.

TIP
You can also manage your custom fields by way of the Portal API (PAPI) or use this API in your scripts for managing custom fields.

For more information about the Portal API, see Portal API (PAPI).

Add Custom Fields

Follow these steps:

1. Log in to API Management SaaS as a Portal Admin.
2. From the menu bar, select the gear icon, **Custom Fields**.
3. Select **Add Custom Field**.
   The Custom Fields page appears.
4. Complete the following fields, and then select **Save**:
   Field Name
   Defines the name for the field you are adding.
   Unique: Yes
   Maximum length : 255
   Required: Yes
   Description
   The description of the field you are adding.
   Maximum length: 255
   Required: No
   Form
   Defines the custom field type (Applications or APIs). Do *one* of the following:
   – To add a custom API field to the API wizards, select **APIs**.
   – To add a custom application field to the Add and Edit Application pages, select **Applications**.
   Input Type
   Defines the input type.
   Options: Text or Single-Select Dropdown
   Required: Yes
   Options
   (If you clicked **Single-Select Dropdown** as the input type) Add one or more options. Remove any blank options. You can drag the options to reorder them.
   Required: Yes
   State
   Defines the state of the custom field.
   Values: Enabled and Disabled
   Default: Disabled
   Required: Yes
   Required Field
   To define the custom field as an optional field, select **No**.
   Required: Yes

NOTE
If you are adding an optional custom application field, add an option such as "None". API developers who choose not to use this custom application field have the option of selecting the "None" option. Ensure that the API Gateway policy processes the "None" option appropriately.

The custom field appears on the **Custom Fields** page in alphabetical order.

Edit Custom Fields

Follow these steps:

1. Select **Administration**, **Custom Fields**.
2. Select **Edit** next to the custom field you want to edit.
   The **Edit Custom Field** page opens.
3. Edit the custom field.
   NOTE
   You cannot change the value for the **Form** or the **Input Type** fields.
4. Select **Save**.

Your changes to the custom field are saved.

Delete Custom Fields

Prerequisite: The custom field is disabled.

Follow these steps:

1. Select **Administration**, **Custom Fields**.
2. Select **Delete** next to the custom field that you want to edit, and then select **Save**.

The custom field is deleted.

Set Up Custom Domain Names

After obtaining Administrator credentials to Portal, which includes getting the access token, the Administrator can change the default URL a tenant has been given to a custom domain name.

This page describes how to:

Set Up a Custom Domain Name

The workflow for setting up your custom domain name is as follows:

Change the Default URL

When you create a Portal tenant, by default, the tenant is provided with a sample domain name for accessing the Portal in the following format: https://tenantID.dev.ca.com.

The Administrator can customize the default domain name by performing these steps:

1. Create a CNAME DNS entry.
2. Get the Access Token.
3. Generate the Certificate Signing Request (CSR).
4. Purchase the certificate with CSR.
5. Upload the certificate.

In the steps, you will need to follow commands where:

- `<PAPI_Portal_TSSG >` is the Portal API URL, for example, `apim-sgg- phoenixproject.dev.ca.com` .
- `<tenant_id>` is your default domain name, for example, `acmeco` .
- `<customdomain>` is the custom domain you want to change it to, for example `portal.acmeco.com` .

Create a CNAME DNS entry

Create a CNAME DNS entry that is pointing the custom domain to the original portal hostname. For example, portal.acmeco.com → acmeco.dev.ca.com.

> **NOTE**
> **Note:** A hard limit to the length of a custom domain name is implemented, default cap is set to 100 characters.

Get the Access Token

The access token is required to generate the CSR, upload the certificate, update the Portal with the Gateway custom domain, and to delete the custom domain.

To get the access token:

1. Log in to the tenant portal as admin user, for example, `acmeco.dev.ca.com` .
2. Go to Portal API. The API Explorer opens in a new window.
3. In the API drop-down list, select `Portal API (tenantid)` , for example `Portal API (acemco)` .
4. Get the API Key, Shared Secret, and Token Endpoint.
5. Use `base64` to encode `API Key:Shared Secret` .
   ```
   echo -n <API KEY>:<Shared Secret> | base64
   ```
   The following is an example:
   ```
   echo -n e3399b3d4b6a4a2da0b9ff8efa0b6394:4ef66c61578a41de8d28280b18724553 | base64
   ```
6. Fetch the authentication access token by running this command:
   ```
   curl '<Token Endpoint>' -H 'Authorization: Basic <value from Step 5>' --data
    'grant_type=client_credentials&scope=OOB' --compressed
   ```
   The following is an example:
   ```
   curl 'https://apim-ssg-phoenixproject.dev.ca.com/auth/oauth/v2/token' -H
    'Authorization: Basic
    ZTMzOTliM2Q0YjZhNGEyZGEwYjlmZjhlZmEwYjYzOTQ6NGVmNjZjNjE1NzhhNDFkZThkMjgyODBiMTg3MjQ1NTM='
    --data 'grant_type=client_credentials&scope=OOB' --compressed
   ```
   If this step is performed correctly, a json payload is returned.
7. Get the value for `access_token` .

Generate the CSR

Generate the Certificate Signing Request (CSR) using the following command:

```
curl -X POST --compressed "https://<PAPI_Portal_TSSG>/<tenant_id>/tools/1.0/
customdomain/csr/<customdomain>" -H "accept: application/x-pem-file" -H "Authorization:
 Bearer <access_token>" -H "Content-Type: application/json" -d "{ \"country_name\":
 \"<Country: US, CA>\", \"state_or_province_name\": \"<State or Province: NY, WA, BC>
\", \"locality_name\": \"<Locality: Boulder, Raleigh, Vancouver>\", \"organization_name
```

```
\": \"<Organization Name: CA Technologies, Inc.>\", \"organizational_unit_name\":
 \"<Organizational Unit Name: IT Department>\", \"common_name\": \"<Common Name:
 test.customdomain.com>\", \"additional_domain_names\": [ \"<List of SAN Names>\" ] }"
```

The command also generates a private key if one does not already exist. Both the CSR and private key are stored in the vault. The following is an example:

```
curl -X POST --compressed "https://apim-ssg-phoenixproject.dev.ca.com/acmeco/
tools/1.0/customdomain/csr/portal.acmeco.com" -H "accept: application/x-pem-file"
 -H "Authorization: Bearer f778f4fa-896c-4f55-969f-5e6a151ee322" -H "Content-Type:
 application/json" -d "{ \"country_name\": \"CA\", \"state_or_province_name\": \"British
 Columbia\", \"locality_name\": \"Vancouver\", \"organization_name\": \"CA Technologies
\", \"organizational_unit_name\": \"Phoenix Team\", \"common_name\": \"portal.acmeco.com
\" }"
```

Purchase the Certificate with CSR

Bring the CSR to a Certificate Authority (a domain registrar) to purchase a certificate.

Upload the Certificate

Upload the certificate using the following command:

```
curl -X POST --compressed "https://<PAPI_Portal_TSSG>/<tenant_id>/tools/1.0/
customdomain/pem/<customdomain>" -H "accept: application/json" -H "Authorization: Bearer
 <access_token>" --data-binary @/path/to/cert
```

The certificate is validated before saving it to the vault. The following is an example:

```
curl -X POST --compressed "https://apim-ssg-phoenixproject.dev.ca.com/acmeco/tools/1.0/
customdomain/pem/portal.acmeco.com" -H "accept: application/json" -H "Authorization:
 Bearer f778f4fa-896c-4f55-969f-5e6a151ee322" --data-binary @/path/to/portal.acmeco.crt
```

Note that it may take up to 48 hours for the change from a default URL to a custom domain name to take place, during which time the new custom domain will not be available for use.

Update Email in Settings

By default, activation links in emails sent (such as registration) use the default URL. To change those links to use your Custom Domain, follow this section.

To update email in settings:

1. In API Portal, select **Portal API**.
2. From the drop-down list, select **Portal API** (tenant name).
3. Select the **Settings** endpoint, select **PUT**.
4. In the **input** field, fill in `CUSTOM_DOMAIN_NAME` .
5. Fill in the **body** field. The following example uses mycustomdomain.com, replace the value with your custom domain.

```
{
 "Uuid": "{{GENERATED_UUID}}",
 "Name": "CUSTOM_DOMAIN_NAME",
 "Value": "mycustomdomain.com"
}
```

NOTE

The UUID is auto-generated. If you need to see the UUID again, select GET call for CUSTOM_DOMAIN_NAME.

Update CORS Assertion

Add the custom domain to the list of valid portal hostnames:

1. Log in to Policy Manager of TSSG.
2. Locate the **Portal Service Preface** policy.
3. Go to **Service**, **Policy** panel. Expand the policy and locate **Process CORS Request** on line 11.
4. Open **Process CORS Request** assertion, select **Add**.
5. Enter the new custom domain, select **OK**.

NOTE
This change needs to be applied every time you upgrade the Portal integration bundle.

The following steps only need to be done once as the change will persist through updates:

1. Locate the **Portal Custom Messages** policy and open **portal-message-received**.
2. Locate the **Process CORS Request** line item and add the custom domain to the CORS hostname list.
3. In **Portal Custom Messages** policy, open **custom-message-received**.
4. Locate the **Process CORS Request** line item. Copy and paste the contents of **portal-message-received** into **custom-message-received**.

Configure SAML SSO

To use Custom Domains with SAML SSO, see the following sections.

NOTE
This feature is not compatible with SAML SSO (old).

OPTIONAL

Set up a New SAML SSO for Custom Domain

1. Log in to the custom domain URL with Portal administrator credentials that are provided by the system (not the SAML SSO credentials).
2. From the custom domain URL, create a new SAML SSO. For more information about creating a new SAML SSO, see Create a SAML SSO Authentication Scheme.
3. Open the authentication scheme and go to **Provider Configuration**.
4. In the Service Provider Configuration section, see the **Assertion Consumer Service (ACS) URL**. The URL is your custom domain name URL.
5. From **Assertion Consumer Service (ACS) URL**, copy the URL (that contains the custom domain URL) and paste into the application callback URL in an identity provider of your choice.

Update Existing SAML SSO

Update an existing SAML SSO for Custom Domain.

1. Log in to the custom domain URL with Portal administrator credentials that are provided by the system (not the SAML SSO credentials).
2. Select the SAML SSO that you want to edit.

3. Go to custom domain URL and set up the SAML SSO from there. For more information about editing an existing SAML SSO, see Edit SAML SSO Configuration.
4. In the existing authentication scheme, go to **Provider Configuration**.
5. In the **Service Provider Configuration** section, update **Assertion Consumer Service (ACS) URL** and **Service provider ID** to your custom domain name URL.
6. Copy your custom domain name URL from **Assertion Consumer Service (ACS) URL** and paste into the application callback URL in an identity provider of your choice.

Delete a Custom Domain Name

If the custom domain is not needed or no longer in use for any reason, it can be deleted. By deleting the custom domain, you are reverted to the default URL (for example, `acmeco.dev.ca.com`) unless there is another custom domain. The Administrator would also remove a custom domain name when the domain expires to prevent the URL from getting more hits. To delete, use the following command:

```
curl -X DELETE --compressed "https://<PAPI_Portal_TSSG>/<tenant_id>/tools/1.0/
customdomain/<customdomain>" -H "accept: application/json" -H "Authorization: Bearer
 <access_token>"
```

The following is an example:

```
curl -X DELETE --compressed "https://apim-ssg-phoenixproject.dev.ca.com/acmeco/
tools/1.0/customdomain/portal.acmeco.com" -H "accept: application/json" -H
 "Authorization: Bearer f778f4fa-896c-4f55-969f-5e6a151ee322"
```

OPTIONAL

Configure Gateway Custom Domain

After configuring Gateway to use your custom domain, you can update the Portal to use the same custom domain.

Update the Gateway to Use the Custom Domain

1. Create a DNS entry for the Gateway pointing at the TSSG load balancer. The steps to do this will vary depending on the DNS provider you use.
2. Generate a CSR in Policy Manager, and then go to a Certificate Authority (a domain registrar) to purchase a certificate. For more information, see "Generate a Certificate Signing Request (CSR)" in Gateway documentation.
3. Import a signed private key. For more information, see "Import a Private Key" in Gateway documentation.
4. Reconfigure the SSL settings for port 8443 by setting the signed certificate as the **Server Private Key** for port 8443. For more information, see "Manage Listen Ports" and "Listen Port Properties" in Gateway documentation.
5. Update the `cluster.hostname` to the new host name of the Gateway in **Manage Cluster-Wide Properties** through Policy Manager. For more information, see "Manage Cluster-Wide Properties" in Gateway documentation.

Update Portal with the New Gateway Custom Domain

Before you perform this step, you must have already updated the Gateway to use the custom domain. To update Portal with the new Gateway custom domain, use the following command:

```
curl -X POST --compressed 'https://<PAPI_Portal_TSSG>/<tenant_id>/tools/1.0/gateway/
portal_gateway_custom_domain' -H "accept: application/json" -H "Authorization:
 Bearer <access_token>" -d "{ \"old_gateway_hostname\": \"<old gateway hostname>\",
 \"new_gateway_hostname\":\"<new gateway hostname>\" }"
```

The following is an example:

```
curl -X POST --compressed 'https://apim-ssg-phoenixproject.dev.ca.com/acmeco/tools/1.0/
gateway/portal_gateway_custom_domain' -H "accept: application/json" -H "Authorization:
 Bearer c4f2a19a-7023-4294-8032-74c1379bacc3" -d "{ \"old_gateway_hostname\": \"acmeco-
ssg.dev.ca.com\", \"new_gateway_hostname\":\"gateway.acmeco.com\" }"
```

Enable Google Analytics Tracking

API Portal administrators can use Google Analytics to track traffic to public Portal pages. To enable this feature, an administrator adds the Google Analytics tracking code to the Settings page. Then the API Portal inserts the tracking code in all public pages.

> **NOTE**
> Google Analytics cannot track the use of private Portal pages, such as the APIs page and Account Plans page. Only API publishers use private Portal pages.

Prerequisites: An API Portal administrator needs a Google Analytics account and tracking code.

To enable Google Analytics tracking:

1. Log in to the API Portal as an **administrator**.
2. From the menu bar, click the gear icon and select **Settings**.
3. Enter the tracking code in the **Google Analytics** field.
4. Select **Save**. Google starts collecting metrics from the Portal within 2 minutes, depending on Google performance.

To disable Google Analytics tracking:

1. Log in to the API Portal as an **administrator**.
2. From the menu bar, click the gear icon and select **Settings**.
3. Remove the tracking code from the **Google Analytics** field.
4. Select **Save**.

Configure Security

This topic describes some settings pertaining to web security and encryption for the API Portal.

Contents:

- Global XSS Filter

Global XSS Filter

The Global XSS (Cross-Site Scripting) filter protects Portal's backend services from malicious scripts. When enabled, user requests (POST, PUT, PATCH) that contain a request body are parsed against a predefined AntiSamy policy. A base policy is provided but a custom one can be configured by the Portal administrator.

Docker Swarm Configuration

Enabling the Feature

The Global XSS filter is disabled by default. You may enable it using the /Settings PAPI endpoint or by executing a PUT request in API Explorer with the **FEATURE_FLAG_REQUEST_XSS_FILTER** feature flag:

```
{
    "Uuid": "<Setting UUID>",
    "Value": "true",
    "Name": "FEATURE_FLAG_REQUEST_XSS_FILTER"
```

```
}
```

Custom Policy Configurations with AntiSamy

The Global XSS Filter uses the OWASP AntiSamy API and policy for malicious script detection. Therefore, the custom policy must follow AntiSamy's format in order for it to work seamlessly. The base policy can be found here. You may make a copy and modify it according to your use case.

Docker Swarm users must ensure that the customized policy is converted to a system usable value. The AntiSamy policy file must be zipped first and then converted to base64. The following is a snippet from the conf.sample that summarizes the process:

```
# ANTISAMY FILTER POLICY
# The value of this variable should contain rules for sanitizing malicious scripts
# that exist in a HTTP request.  The value must be zipped and base64 encoded.
#
# Run the following commands to create the zipped and base64 encoded value:
#   $ zip output.zip your_policy_file.xml
#   $ cat output.zip | base64
# Note: the output from base64 should not contain any line breaks.
# Take the base64 output and set it to the variable below and restart the portal stack.
#
# ANTISAMY_FILTER_POLICY=
```

The following sample configuration snippet shows the processed value assigned to the policy variable:


```
…
ANTISAMY_FILTER_POLICY=UEsDBBQAAAAIACRfLFX9mc/6liYAANMaAQAWABwAdXNlcl9wb2xpY3lfY29uZmlnLnhtbFVUCQADBIEfY4YDIGN1eAsAAQT1
Beq0leTokaTNPdy4bi5J5/NN2nTOuWnvIn8ZiIQk1CTBEqRkNUr/9m93wZdEQKKdxJebr9NJbOK5i33vAnl8ehUGbCESLVV0cvhgeP
+QichTvoxmJ4dn5y8Hf/nLo78OHhyy028PDg4efzYYHPz81VOWZIHQLBFpIsVC+GyaqPD4YJ6m8fFotFwuh8uvhiqZjV79YzRPw+Dr
+w9GOk0yLx3NAjXhwR...
…
```


After the processed value is inserted, save the portal.conf file and restart the Portal for the changes to take effect.

Kubernetes-Helm Chart Configuration

For Kubernetes users interested in learning how the AntiSamy policy is implemented in the Portal Helm Chart, refer to the README in the CAAPIM/apim-charts GitHub repository.

> **NOTE**
> See Also:
>
> - Enable Hashed Client Secret

Enable Hashed Client Secret

The client ID and client secret comprise the OAuth client credentials used for OAuth authorization of non-public apps. The client secret is a shared secret known to the application and the authorization server. Hashing it is recommended for security reasons. One-way hashing of the client ID and client secret provides additional security against attackers is by hiding the plaintext OAuth credential values from view in both the interface and the database.

The client secret initially appears as plaintext after generation. This allows you to copy and provide it to the app developer. If hashing of the client secret is enforced or selected, this initial view is your only opportunity to copy the OAuth credential. For all subsequent visits to the page, the OAuth credential appears as a string of asterisks.

Enable Hashed Client Secret

Enabling hashed client secrets affects newly created applications. It does not impact existing apps.

To configure the format of shared secrets:

1. Log in to the API Portal as an administrator.
2. From the menu bar, select gear icon, **Settings**.
3. In the **Application Shared Secret Security** section, select one or both of the following options:
   – Allow plaintext secrets
   – Allow hashed secrets
4. If you select **Allow hashed secrets**, choose one of the following supported hashing algorithms, and then click **Save**:
   – SHA-512
   – SHA-384
   – SHA-256
   – SHA-1
   – MD5

   NOTE
   Portal does not support HMAC versions.

Consequences of Hashed Client Secrets

Allowing for hashed client secrets results in the following consequences:

- Support for hashed OAuth credentials is introduced with OTK 4.4. You cannot sync applications created with client secret hashing enabled on proxies running versions of OTK 4.3 or earlier. You can sync applications with plaintext client secrets on all versions of OTK.
- Portal records sync failures due to hashed credentials and incompatible OTK version and displays these under the **Deployments** tab of the application. Error messages appear in the sync log.

Generate a New Client Secret

You can always generate a new secret for the application. This can be useful if your shared secret is compromised. When you generate a new shared secret, the API proxy no longer accepts queries that use the old secret. The app developer must update the shared secret in their application before their application can access the APIs.

To generate a new client secret:

1. Go to **Manage, Applications**.
2. Click on the application for which to generate a new client secret. Select **Action > Edit Keys**.
3. In the *Authentication & Keys* section, click on the API key for which to generate a new client secret.
4. Under **Shared Secret (Client Secret)**, click **Generate New Secret**.
5. A new client secret is generated in plaintext. A notification appears, reminding you that generating a new key breaks access for anyone using the previously-issued key.
6. Copy the value and provide both the client ID and new client secret to the app developer.

   IMPORTANT
   If client secret hashing is enabled, this is your only opportunity to copy the value. Do NOT click Save Key before copying the value.
7. If your Portal is configured to allow generation of client secrets in plaintext or hashed format, an option appears to select your preferred type. Select your **Secret Type**.
8. Click **Save Key**.

Audit Logs

To see history of actions that are performed on certain objects, administrators can access audit log files.

> **WARNING**
>
> Audits are guaranteed to be stored for a maximum period of 90 days. Any audits older than 90 days may be deleted. We recommend that you export the audit logs before the expiry period.

The following table shows the objects that are audited for the recorded events:

Object audited	Recorded event
API	Created, Updated, Deleted
Application	Created, Updated, Deleted
Organization	Created, Updated, Deleted
API EULA	Created, Updated, Deleted
API Plan	Created, Updated, Deleted
Account Plan	Created, Updated, Deleted
Custom Field	Created, Updated, Deleted
User	Created, Updated, Deleted, Login Success, Login Failed

Administrators can:

View Audit Logs

As an administrator, you can access data for the audited objects within Portal.

Follow these steps:

1. In the dashboard, select **Administration**.
2. Select **Audit Logs**.
   The Audit Logs page opens. By default, the latest audit logs appear at the top of the list.
3. You can sort by any column except for the **Agent** column:
   - **Time (UTC)**
   - **Object** - Shows the entity for the log entry
   - **Object Name**
   - **Action** - Shows options: **Created**, **Updated**, **Deleted**, **Login Success**, **Login Failed**.
   - **User** - The username of the user who performed the action. A user value **Portal API**, means that the audit event was created through a Portal API call. A user value **Registration Service** shows for users that are not yet registered in Portal.
   - **Agent** - Shows where the action originated (Mozilla, Python, Chrome, and so on), or CA API Gateway.

The following image shows examples of log entries:

Audit Logs

Time (UTC)	Object	Object Name
2018-10-15 20:32	Organization	new org 02
2018-10-15 20:30	Custom Field	oct15 via UI
2018-10-15 20:30	Custom Field	oct15 via papi
2018-10-15 20:25	Custom Field	oct15 via UI
2018-10-15 20:25	User	admin
2018-10-15 20:25	User	auth0|5ab8bf678bd506
2018-10-15 20:25	Organization	new org 02
2018-10-15 18:55	Organization	new org 02
2018-10-15 18:55	User	auth0|5ab8bf678bd506
2018-10-15 18:54	User	orgadmin1

Display: 10

View, Filter, and Sort Audit Logs Using an API Call

Follow these steps:

1. In the dashboard, select **Portal API**.
2. From the **API** drop-down list, select **Portal API**.
   A list of Portal APIs shows.
3. Select **Auditing: View and filter audit logs**.
4. Select the **/tenant-admin/1.0/audits** endpoint.
5. (Optional): Enter the start time (**startTs**) and end time (**endTs**) parameters.
   Note: Enter the UNIX Epoch timestamp in milliseconds. **Example:** 1539707873000.
6. (Optional): To filter the results, enter a value for the parameter field that you want to filter by.
   For example, to see only audit events for the entity type "Application", enter value "Application" in the **entityType** parameter field.
   The following list shows available **entityType** options:

- API
- Application
- Organization
- API EULA
- API Plan
- Account Plan
- User
- Custom Field

7. (Optional): Enter the value for the **size** parameter. The default value is 10. The maximum value is 2000.
8. (Optional): To sort the results, add a custom parameter. See Add a Custom Parameter for /tenant-admin/1.0/audits.
9. Select **Submit**.

A list of audit logs is returned.

Add a Custom Parameter for /tenant-admin/1.0/audits

OPTIONAL

Add a custom parameter to sort results.

Follow these steps:

1. Go to **Portal API**, **Auditing: View and filter audit logs**.
2. Select the **/tenant-admin/1.0/audits** endpoint.
3. Select **Add Parameter**.
4. In the **Parameter** column, enter **sort**.
5. In the **Value** column, enter the value that you want to sort by.
   Example: To sort by **entityType**, enter value **entityType,ASC** for ascending order or **entityType,DESC** for descending order. If no sort field is specified, the default is ASC. The most recent audits are displayed first.
   Note: Sorting by "simplified agent" is not available at this point.
6. Select **Submit**.

A list of audit logs sorted by entityType in ascending or descending order is returned.

Export Audit Logs Using an API Call

Follow these steps:

1. In the dashboard, select **Portal API**.
2. From the **API** drop-down list, select **Portal API**.
   A list of Portal APIs shows.
3. Select **Auditing: View and filter audit logs**.
4. Select the **/tenant-admin/1.0/audits/export** endpoint.
5. Enter the start time (**startTs**) and end time (**endTs**) parameters.
   Note: Enter the UNIX Epoch timestamp in milliseconds. Example: 1539707873000. The maximum range for query data is 90 days.
6. Select the format of the exported document:
   - CSV
   - JSON
7. (Optional): To add custom parameters, see Add a Custom Parameter for /tenant-admin/1.0/audits/export.
   Size and page parameters are not available for this endpoint.
8. Select **Submit**.

A download link with an exported list of audit logs is returned. The export file is .zip regardless of the format selected.

Add a Custom Parameter for /tenant-admin/1.0/audits/ export OPTIONAL

For the Auditing Portal API, you can add custom parameters to filter results.

Following are the available custom parameters to filter by:

- **entityType**
- **entityName**
- **action**
- **userName**

Follow these steps:

1. Go to **Portal API**, **Auditing: View and filter audit logs**.
2. Select the **/tenant-admin/1.0/audits/export** endpoint.
3. Select **Add Parameter**.
4. In the **Parameter** column, enter, for example, **action**.
5. In the **Value** column, enter the value that you want to filter by, for example, **Created**.
6. Select **Submit**.

A list of audit logs filtered by action "Created" is returned.

For more details about APIs for audit logs, see the **Auditing** Swagger JSON file in Portal API (PAPI).

Update the Gateway with New Portal Certificates

The existing wildcard certificate (*.layer7.saas.broadcom.com) for the API Portal is set to expire on November 22, 2022. Updating this certificate ensures that the connectivity between the Gateway and Portal remains intact after new certificates are deployed.

The Gateway Policy Manager is used to perform the majority of the steps outlined in the following procedures:

Retrieve Hostname Values from Cluster Properties

To retrieve hostname values from Gateway cluster properties:

1. Login to the Gateway Policy Manager from your local machine and connect to one of your Gateway instances with Admin access.
2. Go to **Tasks** > **Global Settings** > **Manage Cluster-Wide Properties**
3. Copy the values of the following cluster-wide properties:
   - portal.config.apim.host
   - portal.config.pssg.sync.host
   - portal.config.dssg.datalake.host
4. Click **Close**.

Remove Previous Certificates

To remove previous certificates:

1. In the Policy Manager, go to **Tasks** > **Certificates, Keys and Secrets** > **Manage Certificates**
2. From the list of certifictes, locate the three certificates with the following given names:

  - portal.config.apim.host
  - portal.config.dssg.datalake.host
  - pssg
3. Click **Remove** for each of the three certificates identified in the previous step and confirm their removal.

Add New Certificates

To add a new certificate for 'portal.config.apim.host':

1. In the Policy Manager, go to **Tasks** > **Certificates, Keys and Secrets** > **Manage Certificates**
2. Click **Add**.
   The Add Certificate Wizard opens.
3. On the Enter Certificate Info tab, select the **Retrieve via SSL Connection (HTTPS or LDAPS URL):** option and provide the URL of the portal.config.apim.host by prepending it with “https”. For example: https://{portal.config.apim.host}
4. Click **Next**. A pop-up window appears to confirm your acceptance of the certificate. Click **Accept**.
5. On the View Certificate Details tab, update the **Certificate Name** with the 'portal.config.apim.host' value and click **Next**.
6. On the Specify Certificate Options tab, select the **Outbound SSL Connections** option and click **Next**.
7. On the Configure Validation tab, select the **Certificate is a Trust Anchor** option and click **Finish**.

To add a new certificate for 'portal.config.dssg.datalake.host':

Repeat the same steps outlined for adding a new certificate for 'portal.config.apim.host' except:

- For the **Retrieve via SSL Connection (HTTPS or LDAPS URL):** option, enter https://{portal.config.dssg.datalake.host} as the URL value.
- On the View Certificate Details tab, update the **Certificate Name** with the 'portal.config.dssg.datalake.host' value.

To add a new certificate for 'pssg':

Repeat the same steps outlined for adding a new certificate for 'portal.config.apim.host' except:

- For the **Retrieve via SSL Connection (HTTPS or LDAPS URL):** option, enter https://{portal.config.pssg.sync.host} as the URL value.
- On the View Certificate Details tab, update the **Certificate Name** with the 'pssg' value.

Verify Status of Deployment

After updating the Portal certificates, ensure that the Portal and Gateway remains connected.

To verify your API management asset deployment:

1. Login to the API Portal.
2. Update an API, Application or API key, Account Plan, or API Plan to trigger a deployment. For an On-Demand proxy, trigger the deployment manually.
3. View the Proxy Details page by going to **Publish** > **Proxies** and clicking the name of the proxy.
4. Check the 'Last Updated' field in the appropriate deployment information tile to ensure the deployment activity was captured for the API management asset.

Re-Import Expired APIM Ingress Certificate

Re-Import Expired APIM Ingress Certificate Pre-Update

If you are experiencing issues with outbound connection to Portal stack, re-import your certificate:

1. In CA API Gateway Policy Manager, select **Tasks**, **Certificates, Keys and Secrets**, **Manage Certificates.**
2. Click **Add**.
3. The **Add Certificate Wizard** opens.
4. Select **Import from a file** and click **Browse**. See below for certificates based on your tenant region.
   Locate your certificate and click **Open**.
5. Click **Next**.
6. Select **Outbound SSL Connections.**
7. Click Next.
8. Select **Certificate is a Trust Anchor**.
9. Click **Finish**.

Re-Import Expired APIM Ingress Certificate Post-Update

If you are experiencing issues with outbound connection to Portal stack, re-import your certificate:

1. In CA API Gateway Policy Manager, select **Tasks**, **Global Settings**, **Manage Cluster-Wide Properties**.
2. Locate the **portal.config.apim.host** property.
3. Click **Edit**.
4. Copy the **Value** and click **OK**.
5. Select **Tasks**, **Certificates, Keys and Secrets**, **Manage Certificates.**
6. Click **Add**.
   The **Add Certificate Wizard** opens.
7. Select **Retrieve via SSL Connection (HTTPS or LDAPS URL)**, type in **https://** and paste the value that you copied earlier.
8. Click **Next**.
9. Click **Accept**.
10. Select **Outbound SSL Connection.**
11. Click **Next**.
12. Select **Certificate is a Trust Anchor**.
13. Click **Finish**.

APIM Ingress Certificate

The following certificates are for use with Portal version 4.3.

../../../../../../assets/docops/apiportal/emea_apim_ingress_certificate.crt

EMEA MD5: d333e1b040f0f586c8ef99ecca6991cd

../../../../../../assets/docops/apiportal/apj_apim_ingress_certificate.crt

APJ MD5: 475abb0cbf9752f4dad5e5de7030ea34

Update Portal Integration Software

This topic describes the steps required to update the Portal Integration software, also known as the Portal Integration bundle.

Updating to the latest version of the integration software (also known as the Portal Integration bundle) ensures that API and API key deployment synchronization between the Portal and On-Premise proxies are optimized for reliability and scalability. It also ensures that the API Portal is able to capture and present richer data in the Proxy Details page for analysis and troubleshooting.

When an update for the integration software is available as part of a Portal release, its availability is highlighted in the Release Notes. Portal administrators should coordinate with the API proxy administrator to update the integration software on the API proxy.

See Compatibility Matrix to learn what the latest version of the integration software is and how to identify if you have the latest version installed on your proxy.

WARNING

- The update overwrites any customizations to standard services installed by the Portal integration software, policies, policy templates, or encapsulated assertions. The update does not affect non-standard services, policies, policy templates, or encapsulated assertions. It also does not affect scheduled tasks, or the cached age of APIs and Account Plans (cluster properties).
- This update feature does not update the version of the API proxy. This upgrade feature only upgrades the integration software. For information about general API proxy updates, see Upgrade CA API Gateways in the online documentation for the API Gateway.
- **For customers using API Gateway 10 CR1 and higher:** Download the PortalUpgradeAssertion replacement file to replace the existing server module file when performing your update. For more information, see KB 201757: Upgrade Portal Integration bundle operation fails for API Gateway 10 CR1 and above.

Follow these steps:

1. In the Policy Manager, log in to the API proxy as an administrator.

   NOTE
   If you upgraded your API Portal prior to your attempt in updating the Portal integration bundle, your API proxy may have disconnected. If this is the case, restart the API proxy before proceeding with the integration bundle update:

   ```
   $ service ssg stop
   $ service ssg start
   ```

2. (*For API Gateway 10 CR1 and higher only; skip this step if you are using other versions of the Gateway OR if you have already replaced the PortalUpgradeAssertion file after the Portal 5.0.0 upgrade*) Download and replace the PortalUpgradeAssertion file. Follow the instructions in KB 201757: Upgrade Portal Integration bundle operation fails for API Gateway 10 CR1 and above.
3. On the **Tasks** menu, click **Extensions and Add-Ons**, **Update Portal Integration**.
4. Restart the API proxy. To do this, open a privileged shell on the API proxy and then run these commands:

   ```
   $ service ssg stop
   $ service ssg start
   ```

   For more information, see 'Using the Privileged Shell' in the online documentation for the API Gateway.

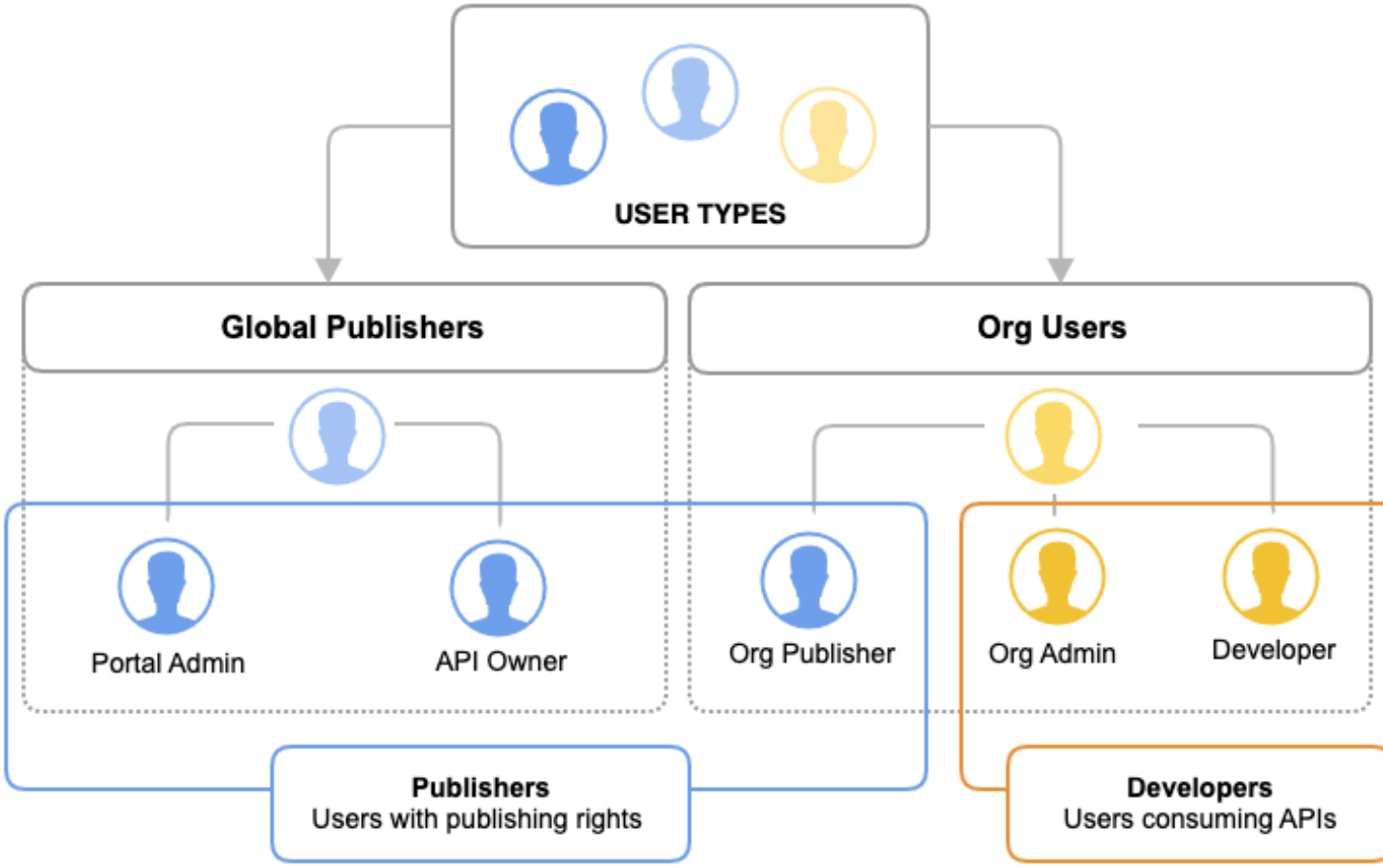
User Types, Roles and Permissions

This article explains the user types that exist in API Management SaaS and what permissions are assigned to each role within the user type.

Users using API Management SaaS can be:

- **Anonymous** users. Anonymous users can access only a few API Management SaaS features, but cannot log into API Portal.
- **Guest** users. Guest users can log in to API Management SaaS, can access a few API Management SaaS features, but do not belong to an organization.
- **Registered** users. Registered users can access more features than anonymous or guest users. Before developers can use the published APIs, they and their developer organization must be registered.
  For more information about registration, see Configure User Registration.

During registration, users are assigned a role. The role determines which features user can access and which tasks they can perform. Two user type categories exist: Global Publisher and Org User.



- The **Global Publishers** category has two roles: Portal Admin and API Owner. These roles are called global because the users are not associated with any organization:
  - The **Portal Admin** can manage API Management SaaS, Portal tenants, APIs, applications, and proxies.
  - The **API Owner** can create APIs and manage those APIs. Users with this role can also manage APIs that belong to other publishers depending on permission settings.
- The **Org Users** category has the following roles:
  - **Org Publisher.** This role is a similar role to the API Owner but users with this role can create and deploy APIs for only the organization to which they are assigned. Users with this role can view analytics across organizations to understand how the APIs that their organization owns is consumed by other organizations.
    For more information about analytics, see Monitor.
  - **Org Admin.** Users with this role are generally developers that can get API keys for the applications that their organization develops.
  - **Developer.** Users with this role develop applications that consume APIs. Depending on the user's permission settings, users with this role can select APIs to be consumed by their application and can choose an API plan for that API's consumption.

For a complete breakdown of user roles and permissions, see the following tables.

> **NOTE**
> Portal Admins can give Org Admins and Developers permission to manage APIs directly assigned to their organization by way of the Authorization API.
>
> **TIP**
> You can also manage your user accounts by way of the Portal API (PAPI) or use this API in your scripts for managing user accounts.

Service Summary

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Manage	✓	✓	✓	✓	✓	✗	✗
Monitor	✓	✓	✓	✓	✓	✗	✗
Administration (gear icon)	✓	✓	✓	✓	✗	✗	✗
Portal API	✓	✗	✗	✗	✗	✗	✗
Content Management (includes setting up Custom Pages)	✓	✗	✗	✗	✗	✗	✗
API Hub (visible if configured)	✓	✓	✓	✓	✓		
Custom Pages (visible if configured)	✓	✓	✓	✓	✓	✗	✗

Manage Permissions

APIs

NOTE
The Org Publisher has publishing permissions only in their assigned organization.

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Add API	✓	✓	✓	✗	✗	✗	✗
Read API	✓	✓	✓	✓	✓	✗	✗
Update API	✓	✓	✓	✗	✗	✗	✗
Delete API	✓	✓	✓	✗	✗	✗	✗
Assign Managing Org	✓	✓	✗	✗	✗	✗	✗
Assign User Permissions	✓	✓	✗	✗	✗	✗	✗
Assign Consuming Org	✓	✓	✓	✗	✗	✗	✗

Apps, Proxies, Plans

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Apps	✓	✓	✓	✓	✓ Read & edit only	✗	✗
API Keys	✓ Can also re-enable non-default keys	✓ Can also re-enable non-default keys	✓	✓ Add & edit only for default key Read only for all	✓ Edit only for default key Read only for all	✗	✗
EULAs	✓	✓	✓	✗	✗	✗	✗
Proxies	✓	✗	✗	✗	✗	✗	✗
Gateway Bundles	✓	✗	✗	✗	✗	✗	✗
API Groups	✓	✓	✓	✗	✗	✗	✗

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
API Explorer	✓	✗	✗	✗	✗	✗	✗
Rate Limits and Quotas	✓	✓ Read only	✓ Read only	✗	✗	✗	✗
Manage API Plans	✓	✓	✗	✗	✗	✗	✗
Select API Plan for use in Application	✓	✓	✓	✓	✓	✗	✗

Administration

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Users	✓	✓ Read only	✗	✓	✗	✗	✗
Organizations	✓	✓ Read only	✓ Read only	✗	✗	✗	✗
Basic Settings	✓	✗	✗	✗	✗	✗	✗
Registration	✓	✗	✗	✗	✗	✗	✗
Requests	✓	✓	✓	✗	✗	✗	✗
Request Settings	✓	✗	✗	✗	✗	✗	✗
Authentication	✓	✗	✗	✗	✗	✗	✗
Custom Fields	✓	✓	✗	✗	✗	✗	✗
Mobile API Gateway	✓	✗	✗	✗	✗	✗	✗
Email Settings	✓	✗	✗	✗	✗	✗	✗
Audit Logs	✓	✗	✗	✗	✗	✗	✗

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Register Customized API Hub	✓	✗	✗	✗	✗	✗	✗

My Profile Permissions

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
	✓	✓	✓	✓	✓	✓	✗

Appearance

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Manage Global Themes	✓	✗	✗	✗	✗	✗	✗

Analytics Permissions

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Monitor	✓	✓	✓	✓	✓	✗	✗

Portal API Permissions

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Authorization, Portal, Portal Metrics, Login (visible only if configured)	✓	✗	✗	✗	✗	✗	✗

Content Management Permissions

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Custom Pages (visible if configured)	✓	✓	✓	✗	✗	✗	✗

Reset Password

Select the drop-down list next to your user role in the top right corner and select **My Profile**.

Manage Users

This article describes how Portal Admins and Org Admins add and edit user account information.

Add Users

Portal Admins and Org Admins can add other users to API Portal. Portal Admins can add users and can assign a role to them. Org Admins can add Developers and other Org Admins to their organization.

> **NOTE**
> Portal Admins can add users only if single sign-on (SSO) is disabled.

If Third-Party Registration is enabled, anonymous users can register themselves and their organization. API Portal automatically assigns these users the Org Admin role. Users cannot add themselves to an existing organization.

If the Registration Request Workflow for Third-Party Registration is also enabled:

1. The user completes the registration form.
2. The Portal Admin approves the registration request.
3. The user can complete the account setup form.

> **NOTE**
> Account plans are associated with organizations.

Prerequisite: The organizations have been added.

Follow these steps:

1. Log in to API Portal as a **Portal Admin** or **Org Admin**.
2. From the menu bar, select the gear icon, **Users**.
   The **Users** page opens.
3. Select **Add User**.
   The **Add User** page opens.
4. In the **User Type** section, select the user type:
   – To add a Portal Admin or API Owner, select **Global Publishers**.
   – To add an Org Publisher, Org Admin, or Developer, select **Org Users**.
5. Add the user details.
6. (For Global Publishers only) Select the role that is applicable to this user, either **Admin** or **API Owner**.
   (For Org Users only) Click **Next** to proceed to selecting one or more organizations and assigning a corresponding role to the user.
   You can filter organizations using the **All Organization Types** drop-down list. A **Publisher** organization type can include **Org Publisher, Org Admin,** or **Developer** role users whereas a **Consumer** organization type can include only **Org Admin** and **Developer** role users.
   > **NOTE**
   > You must assign the user to at least one organization before you can save the user details.
7. Select **Create**.
8. If you left the **Notify User** checkbox selected on the Add User page, API Portal sends an account activation invitation to the user by email. The Users page shows that the user account is pending activation, whether the invitation was sent or not.
   If you cleared the **Notify User** checkbox, then later when you want to send the invitation to the user, complete the following:
   a. Go to the **Users** page.
   b. On the **Actions** menu column for that user, select **Resend Activation Email**.

   > **NOTE**
   > The user account is active after the user accepts the invitation and completes the account setup form.

Search for a User by Name and Email

Consider the following points:

- Search by first name, last name, or full name (Format: First name Last name).
- The **Name** and **Email** fields are self-predictive fields, which means that when you enter at least three consecutive characters, the system fetches the matching records. For example, if you are searching for Robert, then entering "Rob" or "ber" in the Name field shows you the names of all the users containing the search string. This search is case insensitive.
- Wildcard characters are not allowed in the search string. If entered, these are treated as normal characters.
- For email search, the special characters `!#$%&*+/=?^_@.`{|}~-` are allowed.

Filter Your Search Results

From the user list or your search results, you may apply specific filters by a specific organization, role or status:

- **Organization**: Select an organization to display users belonging to that organization. Default is ‘All Organizations’.
- **Role**: Select the role(s) to display users having those roles (e.g., API Owner, Organization Admin, Organization Publisher, Developer)
- **Status**: Select a status to users that have been assigned that status only (e.g., Enabled, Disabled, Locked, Pending Activation, Pending Approval)

Edit User Account Information

You can edit user accounts using the Edit User page. The following rules are applicable:

- You cannot edit your own account.
- (For Publishers only) You cannot edit user details of an external IdP user.
- Portal Admins can:
  - Edit the users who are managed in API Portal. You can map only Developers to multiple organizations and roles.
  - Edit the organization and role of Developers (IdP users) who are authorized using API Portal.

    NOTE
    Portal Admins can manage users using API Portal only if single sign-on (SSO) is disabled.
- The Org Admin can only edit the users who are created and managed in API Portal. The user must belong to same organization as the Org Admin.

You can edit the following account information of a user, added and managed from API Portal:

- Names (but not the username)
- Email address

  NOTE
  Only Portal Admins can change the email address after authenticating with the login password. All the other users such as API Owners, Org users, developers cannot update their own email address.
- Language (but this setting has no effect)
- (For Publishers only) Role (only to another role in the same role category)
- State (enabled or disabled)
- (For Developers only) Organization and role (map with multiple organizations and corresponding roles)

You can edit the following account information of an external IdP with the authorization type as "Portal":

- Organization and role (map with multiple organizations and corresponding roles)

Follow these steps to edit a user added and managed in API Portal:

1. Log in to the API Portal as a Portal Admin.
2. From the menu bar, select the gear icon, **Users**.
3. Click on the user name to edit the user details.
   The **User Details** page appears.
4. Edit the user information.

5. (For Developers only) Select **Next**. Select one or more organizations and a corresponding role to assign to this user.

   NOTE
   You cannot save the user details without assigning the user to at least one organization.
6. Select **Save**.

(Developers only) Follow these steps to edit a user using external authentication scheme to log in to API Portal:

WARNING
To edit an external IdP user from API Portal, you must edit the authentication scheme of this user to change the authorization type to "Portal". But after editing the authentication scheme, all new *Publishers* of this authentication scheme are unable to log in to API Portal. To address this limitation, create an authentication scheme for all the new Publishers. This does not impact the users who have previously logged in to API Portal.

1. While logged in to the API Portal as a Portal Admin, edit the authentication scheme that the user uses:
   a. From the menu bar, select the gear icon, **Authentication**.
   b. Select **Edit** from the **Actions** menu of the authentication scheme.
   c. Go to the **Attribute Mapping** section, select **Portal** from **Select Authorization Type**, and then save the authentication scheme.
2. Map the user to multiple organizations:
   a. Select **Users**.
   b. Go to the **Developers** tab.
   c. Click on the user name to edit the user details.
      The user details are displayed. This page is read-only.
   d. Select **Next**.
   e. From the **Select Organization and Role** page, select the organization and the corresponding role, and then select **Save** to save the mapping.
      The user is mapped to one or more organizations.

Enable and Disable User Accounts

You can enable and disable user accounts. You cannot enable or disable your own account.

Follow these steps:

1. Log in to the API Portal as a Portal Admin or Org Admin.
2. From the menu bar, select the gear icon, **Users**.
3. Click the name of the user that you want to edit.
4. Change the state to Enabled or Disabled, and then select **Save**.

Manage Organizations

This article describes how to register for an organization and how to manage the organization mapping of a Developer.

As a Portal Admin, you can add, view, and can edit organizations in API Management SaaS. Organizations are a way to group and manage related Developers. API Owners can view the list of organizations. When you register a Developer, you assign the Developer to one or more organizations. Before you can register a Developer to a new organization, add the organization to API Portal.

TIP
You can also manage your organizations using the Portal API (PAPI) or use this API in your scripts for managing organizations.

Organization Registration

In API Management SaaS, you group registered application developers into organizations. Organizations contains one or more Developers. Commonly an organization includes Developers working together on one or more applications.

API Portal assigns each Developer to a Developer user account or an Org Admin user account. Org Admin accounts provide more privileges than Developer accounts. API Portal assigns the first registered member of an organization to the Org Admin account.

API Portal assigns Developers to user accounts in the following circumstances:

- When the Developer registers their organization and themselves simultaneously, API Portal assigns the Developer to an Org Admin account.
  For example, if Sharon registers her organization, Lion Systems, and herself on API Portal, then API Portal assigns Sharon to an Org Admin account for Lion Systems.
- The Org Admin sends a registration invitation to the developer. When the Developer accepts the invitation, the API Portal assigns the developer to a Developer user account and the Developer becomes a member of the organization.
  For example, Sharon can have the API Portal send Greg and Chloe invitations to register on the API Portal as members of her organization, Lion Systems. Sharon can specify that she wants API Portal to assign Greg to an Org Admin account and Chloe to a Developer account.

Register your Organization and Yourself

1. From the menu bar, select the gear icon, **Registration**.
2. Complete the Registration form. If you do not enter a name for your organization, API Portal assigns one.
3. Select **Register Now**.
   (If single sign-on (SSO) is not enabled) API Portal sends you an account activation email.
4. In the email message, select the account activation link.
   In API Portal, the Account Setup form opens.
5. Complete the Account Setup form, and then select **Activate Account**.

Your organization is registered. API Portal assigns you to an Org Admin account.

Send a Registration Invitation to a Developer

Prerequisite: Single sign-on (SSO) is not enabled.

1. Log in to API Portal as an Org Admin.
2. From the menu bar, select the gear icon, **Registration**.
3. Select **Invite User**.
   The Invitation form opens.
4. Complete the Invitation form, specifying which Developers get which account types, and then select **Invite**.
   API Portal sends the Developer invitations to register on API Portal by way of an email. When the Developer accepts the registration invitation, the Developer becomes a member of the organization.

Multi-Organization Mapping

You can map a user with a Developer account to multiple organizations, and specify what role this user will have within each organization.

> **WARNING**
> If you are using multiple organizations, PAPI might not work properly because permissions are limited to only one organization at a time.

Which users can be mapped to multiple organizations?

- Developers who are added and managed in API Portal.
- Developers who log in to API Portal using an external authentication scheme, and the authorization type is set to "Portal".

  NOTE
  What is an *authorization type*?

  Depending on how the user account exists, API Portal users can be categorized into two groups:

  - **Portal:** The user details and access levels can be edited and managed in API Portal.
  - **Identity Provider:** The user details cannot be edited and managed in API Portal. However, the user can be assigned to multiple organizations.
    For more information, see Map IdP Users to Multiple Organizations.

Who can map the Developers to multiple organizations?

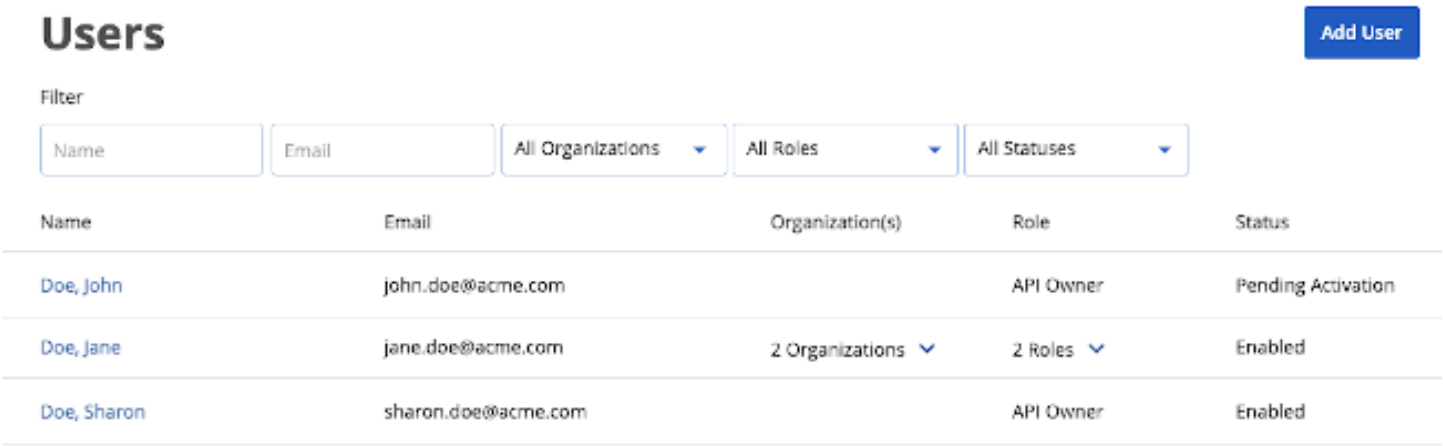
- Portal Admins

How can you map the Developers to multiple organizations?

- While adding a Developer in API Portal.
- (If the user is added and managed in API Portal) While editing a Developer in API Portal.
- (If the user logs in to API Portal using an external authentication scheme) Set the authorization type to "Portal", and edit the Developer details.

How do the organization names appear on the Users page?

Users who are assigned to *multiple* organizations are displayed in the **Users > Org Users** section, in the following format.



You can select **Multiple** to view and edit the organization and corresponding roles.

Add Organizations to API Portal

Portal Admins can add organizations to API Portal using the **Add Organization** form. Every organization requires a name anda rate limit and quota assigned at the organization level. You limit how much the applications developed by a Developer can use the API by way of a rate limit and quota. You can also use rate limits and quotas to give some organizations access to private APIs.

For information about rate limits and quotas, see Manage Rate Limits and Quotas.

Follow these steps:

1. From the menu bar, select the gear icon, **Organizations**.

The Organizations page appears.
2. Select **Add Organization**.
3. Enter a unique name for the organization (up to 255 characters).
4. Select the organization type for the organization. By default, **Consumer** organization type is selected.
   - **Publisher:** Defines an organization that can have Org Publisher role users and allows managing APIs, publishing APIs, and proxy mappings. For example, a Publisher can be an internal organization that publishes an API and also consumes the API from API Portal.
   - **Consumer:** Defines an organization that can have users (Org Admin or Developer) who can only view and consume APIs. For example, a Consumer can be a partner organization that consumes an API from API Portal.
5. Select a rate limit and quota for the organization, enter a public description of the organization, and then click **Save**.

View your List of Organizations

Portal Admins and API Owners can view a list of organizations on the **Organizations** page. The following information is displayed:

- The name of the organization
- Organization tag(s)
- The rate limit and quota for the organization
- The type of the organization
- The status of the organization
  - **Enabled** (Organization is active and contains developers)
  - **Registration Init** (Organization has been registered and is awaiting account setup completion)
  - **Registration Pending Approval** (Organization has been registered, account setup completed, and is awaiting approval)

Edit Organizations

Follow these steps:

1. From the Organizations page, click on the organization name to edit organization details.
   The Edit Organization page appears.
2. Edit the organization details. You can edit the organization name, organization type, the rate limit and quota, and the public description.
   Conversion of the Consumer organization type to a Publisher organization type does not require any conditions but for conversion of a Publisher organization type to a Consumer organization type, the organization must meet the following criteria. The organization should not:
   - have any Org Publisher role users
   - act as managing organization for any API
   - be part of the organization assignment list in any proxy

   NOTE
   Changing the **Organization Type** supersedes user permissions. For example, even if an API Developer has CRUD permissions for an API, the API Developer will be able to perform the CRUD operations only if the **Organization Type** of the user is **Publisher.**
3. Click **Save** after making your changes.

View List of APIs at an Organization Level

Portal Administrators and API Owners can view the list of APIs along with its visibility status at an Organization level.

From the Organizations page, click on the organization name to access the Organization Details page. Click the API Access tab. All the APIs associated with the selected Organization are listed on this page with the following information:

- The name of the API
- Tags associated with the API
- Status of the API (New/Incomplete/Unpublished/Enabled/Disabled/Deprecated)
- The access type of the API (Public/Private)
- The Access column showing if the Organization has access to the API
- The rate limit and quota (API per organization) assigned to the API

You can filter APIs on the API Access page by **API Name**, **Tags**, or **Access Type**. You can toggle the list using the **All, Access, No Access** options.

By default, an Organization has access to all Public APIs. Public APIs are always accessible and you cannot remove access to these APIs.

Private APIs with Managing Organization as the selected Organization have default access and these APIs are always accessible and cannot be removed access from Organizations. These APIs are represented with the text, **Managed**, in the Portal UI.

You can provide the Organization access to Private APIs if needed. To add or remove access for Private APIs, select the checkbox for the API(s) and click **Add Access/Remove Access** depending on your action required. Alternatively, you may click the toggle in the Access column to add or remove access (a checkmark in the toggle indicates that the organization has access to the API).

Revoke Developer Access From an Organization

Org Admins can *only* revoke the access from Developers to their organizations. They cannot delete Developers from organizations.

Follow these steps:

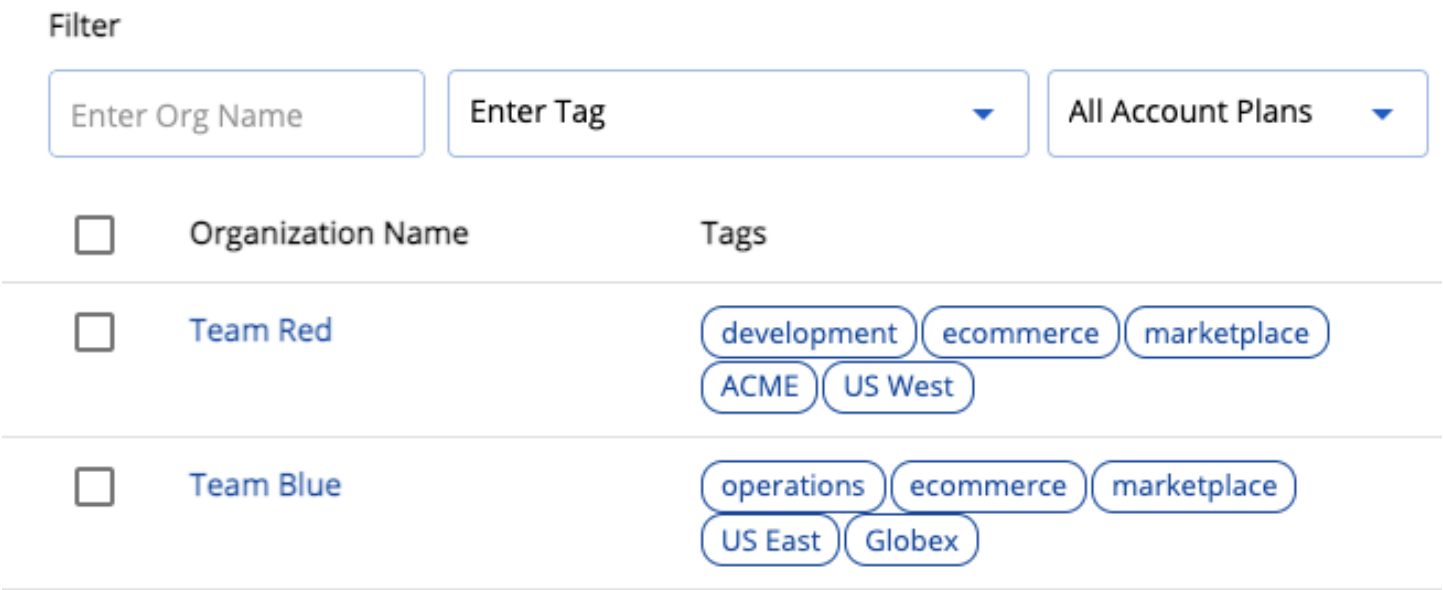
1. Log in to API Portal as an Org Admin.
2. From the menu bar, select the gear icon, **Users**.
3. In the **Actions** menu for the user, select **Revoke Access**.
4. Select **Ok**.

Add Organization Tags

The Portal supports organization tagging to help Portal Admins group related organizations. Common grouping use cases include:

- Internal organizations within a division
- Hierarchy of organizations that consist of divisions and sub-divisions
- Partner organizations from different regions

You may add and delete one or more tags for each individual organization OR bulk add and delete tags to multiple organizations at the same time.



> **NOTE**
> Organization tags cannot be applied to APIs like API tags and vice versa.

Follow these steps:

1. Log in to API Portal as an Org Admin.
2. From the menu bar, select the gear icon, **Organizations**.
3. On the Manage Organizations screen, select a check box for one or more of the organizations you would like to add tag(s) to. To select all the organizations in the list, click the bulk check box at the top of the list.
   The Actions box appears at the top of the screen.
4. In the Actions box, enter one or more new tags or select from a list of existing tags to apply to your organization(s).
5. Click **Add Tag**.
   Alternatively, if you want to remove the selected tag(s), click **Delete Tag**.

Performing a Filter Search by Organization Tags

To perform a filter search of organizations based on tags, click the Enter Tag drop-down list and select one or more existing tags. If you are searching by a single tag, the list displays organizations that have that tag. If you are searching by multiple tags, the list displays organizations with either the first tag, second tag, OR subsequent tags.

Managing Organization Tags with PAPI

You can also manage organization tags using the new `/tenant/-admin/1.0/tags` endpoint in the Portal API (PAPI).

> **NOTE**
> **Using the PATCH Operation**
>
> The operation: `PATCH /tenant-admin/1.0/tags/{tagUuid}/organizations` lets you bulk-update tag-organization associations by updating tags that you've listed in {tagUuid} AND simultaneously deleting the tags you did NOT list as part of the update. This operation allows you to quickly make sweeping tag-organization changes without having to perform a second removal step.

See the latest PAPI swagger file to learn more.

FAQs

Q: I am an Org Admin. Can I remove a user from API Portal?

> No, you do not have the rights to remove a user from API Portal. Only Portal Admins can remove users. Org Admins can only revoke the user access from the organization.

Q: I am an Org Admin. Can I revoke access of any user?

> You can only revoke access of a Developer who is,
>
> - added and managed in API Portal, and belongs to your organization
> - external IdP user with the authorization type changed to "Portal", and belongs to your organization.

Q: I mapped a user to an invalid organization in the external IdP. What happens now?

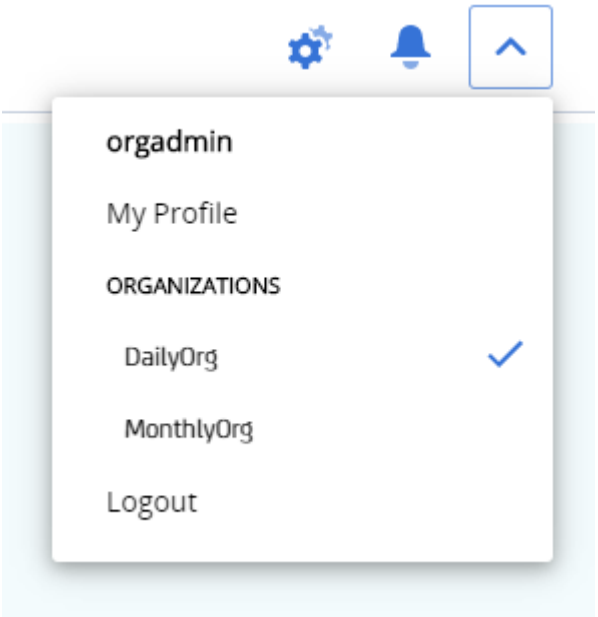
> A user who does not have any access in API Portal or mapped to invalid organization or role while being authorized by IdP, becomes 'an *external user who does not belong to any organization*'. This user may be authenticated using external IdPs or using API Portal. In any case, they are listed in **Users** page with **Organization** and **Role** as 'None'.
>
> To rectify:
>
> - If this user is managed by API Portal, you can edit the details to belong to one or more organizations.
> - If this user is authenticated using external IdPs, you can change the authorization type to 'Portal' and then edit the organization details.
> For more information, see Map IdP Users to Multiple Organizations.

Q: I revoked the user access from all their organizations. Now this user does not belong to any organization. Can this user still use API Portal?

> The user can still log in to API Portal, but can only access the Home page and user profile page. The user is listed in the Users section with organization and role as 'None'. You can map this user to an organization and role, by selecting **Edit User** from the **Actions** menu.

Q: I belong to two organizations, org1 and org2. I am currently logged in to API Portal for organization org1. How do I switch to the other organization? Do I need to log out?

> No, you do not need to log out from API Portal. To use API Portal for your other organization, simply use the switch feature from My Profile, as shown in the following screenshot:


orgadmin
My Profile
ORGANIZATIONS
DailyOrg
MonthlyOrg
Logout

API Portal Dashboard

A quick start guide to using the API Portal dashboard.

This section contains information on the Layer7 API Developer Portal (API Portal) dashboard, navigation, and menu bar.

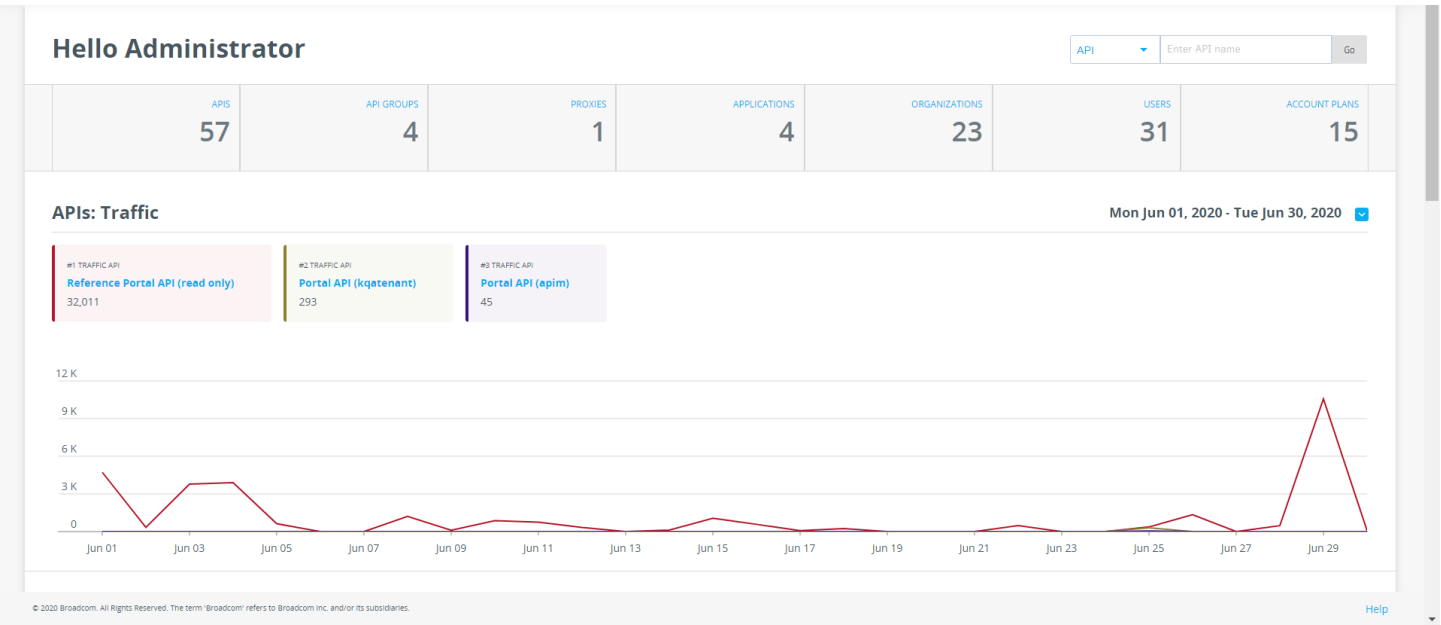
Dashboard

The API Portal Dashboard provides a quick overview of the current state of your API Portal. Upon logging into API Portal, you see the Dashboard with the following information based on your role:

- **Personalized Header**
  Shows a count of the following entities. Click the entities to view detailed information.
  - APIs
  - API Groups
  - Proxies
  - Applications
  - Organizations
  - Users
  - Rate Limits and Quotas
  - API Plans
- **Search Bar**
  Helps you search for APIs, applications, or organizations. Organization Users can search on APIs and Applications only.
- **Analytics**
  Shows the analytics charts for Top 3 APIs traffic, Top 3 Organizations traffic, Top 3 Proxies traffic, and Top 3 Applications traffic with data from the last 30 days. For more information, see the section **Charts** below.

  > **NOTE**
  > If Analytics is disabled, you can only see the APIs list.

Here's a sample dashboard for a Portal Admin:



Charts

Org Admins and Developers can view charts for APIs and applications. Org Publishers can view Proxies chart and also Proxies that are visible to their respective Organization. Only Portal Admins and Publishers can view the organization traffic chart.

Click the drop-down arrow icon beside the date range label to see more options, such as **Chart** and **Move/Collapse Panel**. You can use these options to change the Analytics view as per your requirements. These view preferences are saved and you can view the charts with these preferences the next time you access the Dashboard.

The charts show legend such as APIs traffic and recently accessed pages. Hover on the traffic legend to highlight the respective line on the chart. Click the API legend to navigate to the respective API page.

Role-based Dashboard View

Publishers and Portal Admins can view the information and quickly access all the entities. Org Users can view and access the following entities:

Entities	Org Admin	Developer	Org Publisher
APIs	✓	✓	✓
API Groups	✗	✗	✗
Proxies	✗	✗	✗
Applications	✓	✓	✓
Organizations	✗	✗	✓
Users	✓	✗	✗

Entities	Org Admin	Developer	Org Publisher
Rate Limits and Quotas	✗	✗	✓
API Plans (if enabled)	✓	✓	✓
Proxy Error List	✓	✗	✗

NOTE
Org Publishers can also view EULAs from the header tab.

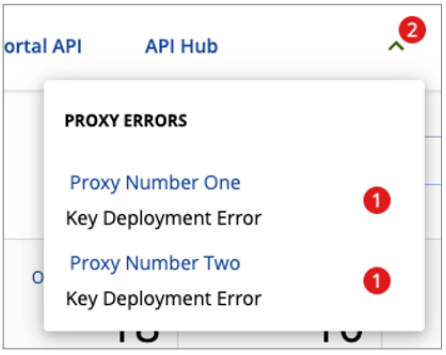
Role-based Menu Options

The top menu bar contains centralized navigation for all task options and access points available to your particular role.

NOTE
For more information on user roles and permissions, see User Types, Roles and Permissions.

- **Manage:** Easily manage entities such as APIs, Applications, EULAs, Proxies, Rate Limits and Quotas, API Plans, and Gateway Bundles. See Manage.
- **Monitor:** Monitor analytics and generate reports for traffic and quota consumption. See Monitor.
- **Portal API:** Opens the API Explorer to gain access to Portal APIs.

  NOTE
  API Explorer is only accessible through the API Portal/Ingress tenant.
- **API Hub:** Directs you to the API Hub homepage. If you have registered other remotely hosted customized API Hub, the customized API Hub displays under this menu item.
- **Proxy Error List:** Available or Org Admin users only. A drop-down menu that summarizes currently known API-proxy related errors, ranging from deployment to connection errors. A red marker on the menu indicates X number of proxy errors, while a green checkmark on the menu indicates that no proxy issues have been found. Each error alert contains a link that you can click to directly access the affected proxy to investigate the error in greater detail. The Proxy Errors list may also display a summary of the number of trusted certificates that have expired or are about to expire within 60 days.



IMPORTANT
To be able to use the Proxy Errors list in the API Portal Dashboard, you must install the latest update of the Portal Integration Bundle on the API proxy. For the latest version of the bundle supported by the Portal, see Compatibility Matrix.

NOTE
The display of expired or expiring trusted certificates is currently an experimental feature and is dependent on the experimental Graphman Assertion. See Progressive Delivery of APIM Features to learn more about the experimental feature category and others.

- **Administration (gear icon):** Manage administration options such as Appearance, Audit Logs, Authentication, Custom Fields, Email/SMTP Settings, Integration, Organizations, Registration, Request Settings, Requests, Settings, and Users.
- **Notifications (bell icon):** View and approve requests and registrations.
- **Account (user icon):** Manage your profile and account. Org users can also use this access point to switch between workspaces or organizations.

Manage

This section includes information about the various tasks Publishers (users with publishing rights) and Developers (users consuming APIs) can do in API Management SaaS.

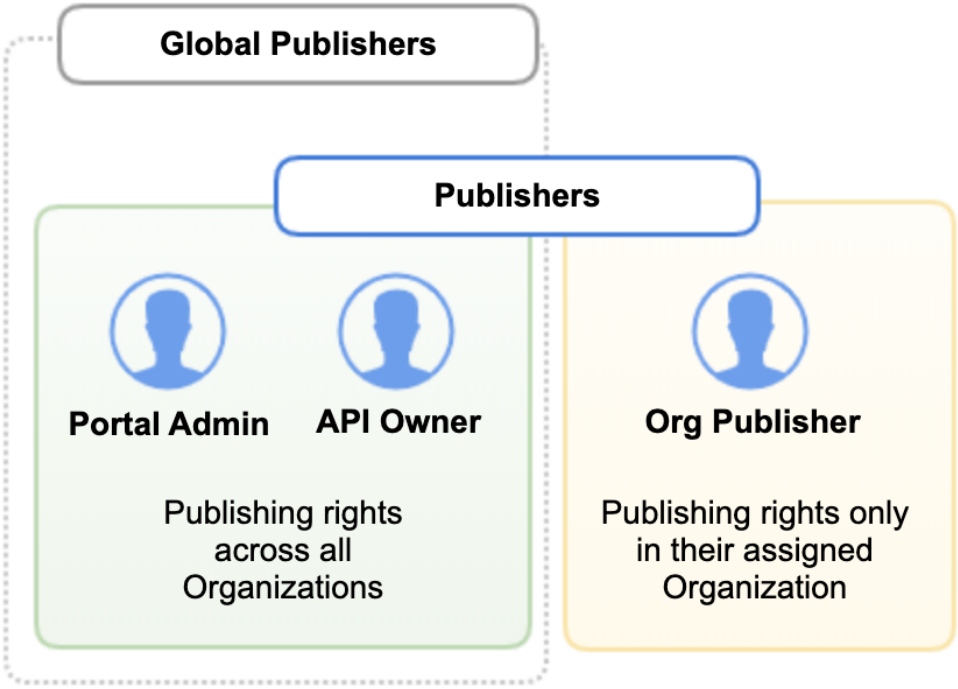
For more information about the user types, see User Types, Roles and Permissions.

In this article:

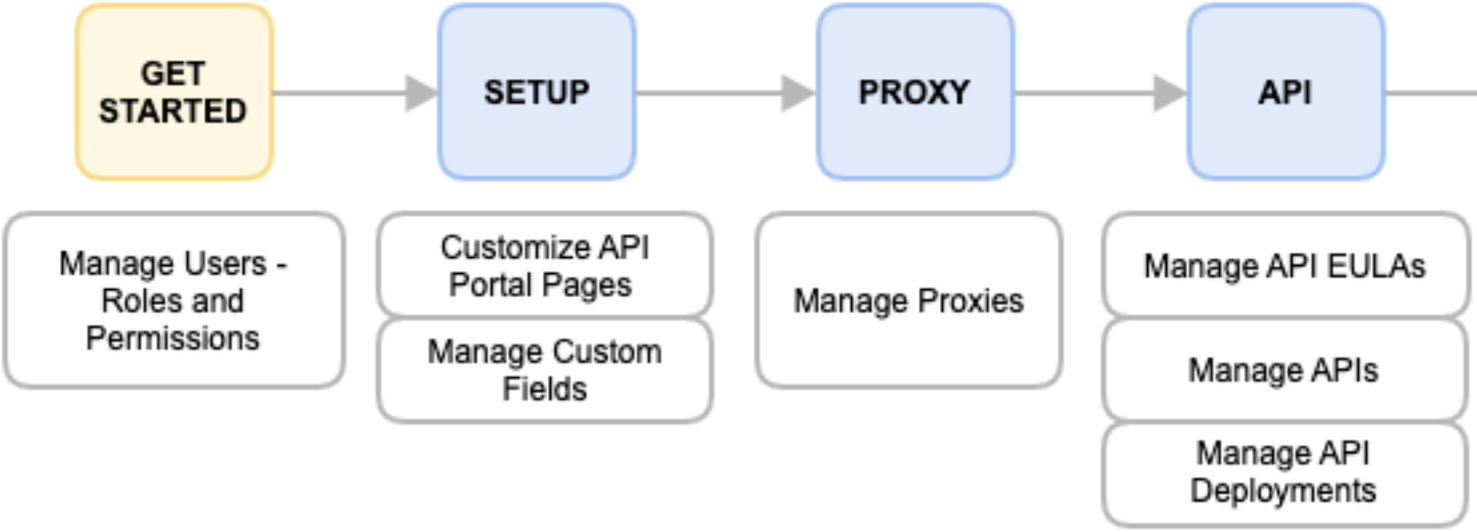
Publishers

The tasks that Publishers can do is based on their role, either Portal Admin or an API Owner. Org Publishers have publishing rights similar to Portal Admins and API Owners but only in their assigned organization.

The following image illustrates the rights for Publishers:



The Publisher tasks in the following diagram are sorted according to the general workflow for setting up and managing a new API Management SaaS:



Publisher Permissions

The following table shows the tasks that Publishers have permission to perform:

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Add API	✓	✓	✓	✗	✗	✗	✗
Read API	✓	✓	✓	✓	✓	✗	✗
Update API	✓	✓	✓	✗	✗	✗	✗
Delete API	✓	✓	✓	✗	✗	✗	✗
Assign Managing Org	✓	✓	✗	✗	✗	✗	✗
Assign User Permissions	✓	✓	✗	✗	✗	✗	✗

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Assign Consuming Org	✓	✓	✓	✗	✗	✗	✗

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Apps	✓	✓	✓	✓	✓ Read & edit only	✗	✗
API Keys	✓ Can also re-enable non-default keys	✓ Can also re-enable non-default keys	✓	✓ Add & edit only for default key Read only for all	✓ Edit only for default key Read only for all	✗	✗
EULAs	✓	✓	✓	✗	✗	✗	✗
Proxies	✓	✗	✗	✗	✗	✗	✗
Gateway Bundles	✓	✗	✗	✗	✗	✗	✗
API Groups	✓	✓	✓	✗	✗	✗	✗
API Explorer	✓	✗	✗	✗	✗	✗	✗
Rate Limits and Quotas	✓	✓ Read only	✓ Read only	✗	✗	✗	✗
Manage API Plans	✓	✓	✗	✗	✗	✗	✗
Select API Plan for use in Application	✓	✓	✓	✓	✓	✗	✗

NOTE
The Org Publisher has publishing permissions only in their assigned organization.

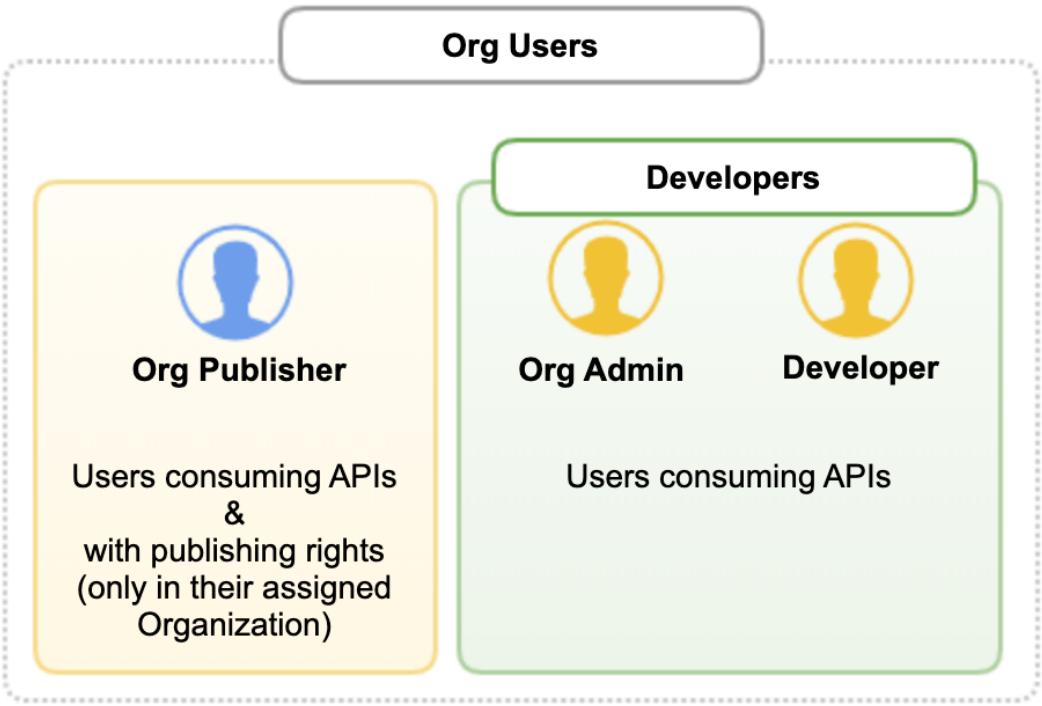
Developers

Developers can:

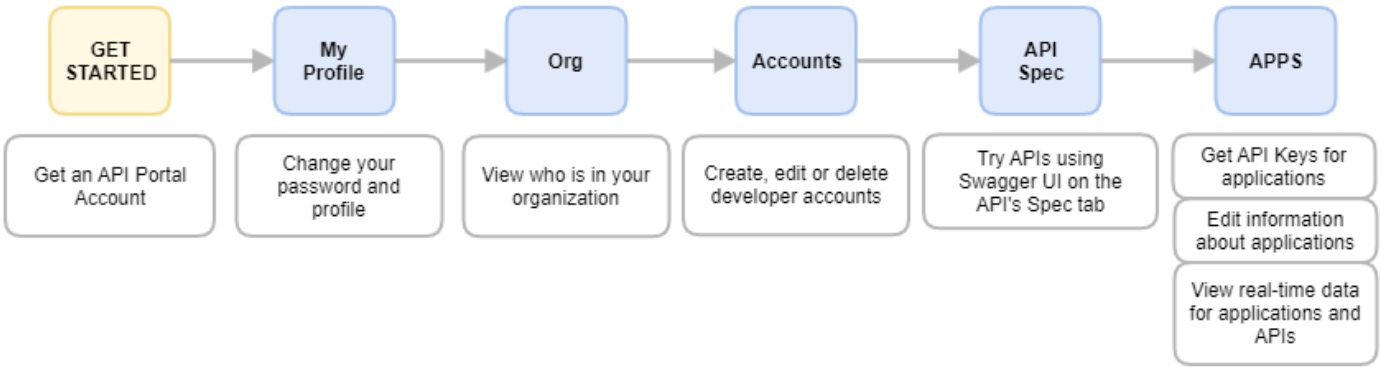
- Explore APIs.
- Get API keys for their applications (Org Admins only).
- View real-time data.
- Display the Proxy URL for the API.

Org Publishers have permission to deploy APIs for an organization. Here the word "organization" refers to a group of developers.

The following image illustrates the rights for Developers:



The Developer tasks in the following diagram are sorted according to the general workflow for setting up and managing a new API Management SaaS:



Developer Permissions

The following table shows the tasks that Developers have permission to perform:

NOTE
Portal Admins can give Org Admins and Developers permission to manage APIs directly assigned to their organization. This permission does not apply to public APIs with no organization assigned or those assigned to other organizations, or private APIs assigned to other organizations through an account plan.

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Add API	✓	✓	✓	✗	✗	✗	✗
Read API	✓	✓	✓	✓	✓	✗	✗
Update API	✓	✓	✓	✗	✗	✗	✗
Delete API	✓	✓	✓	✗	✗	✗	✗
Assign Managing Org	✓	✓	✗	✗	✗	✗	✗
Assign User Permissions	✓	✓	✗	✗	✗	✗	✗
Assign Consuming Org	✓	✓	✓	✗	✗	✗	✗

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Apps	✓	✓	✓	✓	✓ Read & edit only	✗	✗
API Keys	✓ Can also re-enable non-default keys	✓ Can also re-enable non-default keys	✓	✓ Add & edit only for default key Read only for all	✓ Edit only for default key Read only for all	✗	✗
EULAs	✓	✓	✓	✗	✗	✗	✗
Proxies	✓	✗	✗	✗	✗	✗	✗
Gateway Bundles	✓	✗	✗	✗	✗	✗	✗
API Groups	✓	✓	✓	✗	✗	✗	✗
API Explorer	✓	✗	✗	✗	✗	✗	✗

	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Rate Limits and Quotas	✓	✓ Read only	✓ Read only	✗	✗	✗	✗
Manage API Plans	✓	✓	✗	✗	✗	✗	✗
Select API Plan for use in Application	✓	✓	✓	✓	✓	✗	✗

NOTE
The Org Publisher has publishing permissions only in their assigned organization.

Manage APIs

You can publish and manage APIs using the API Portal and API proxy. You can get information about the published APIs using the APIs page.

NOTE

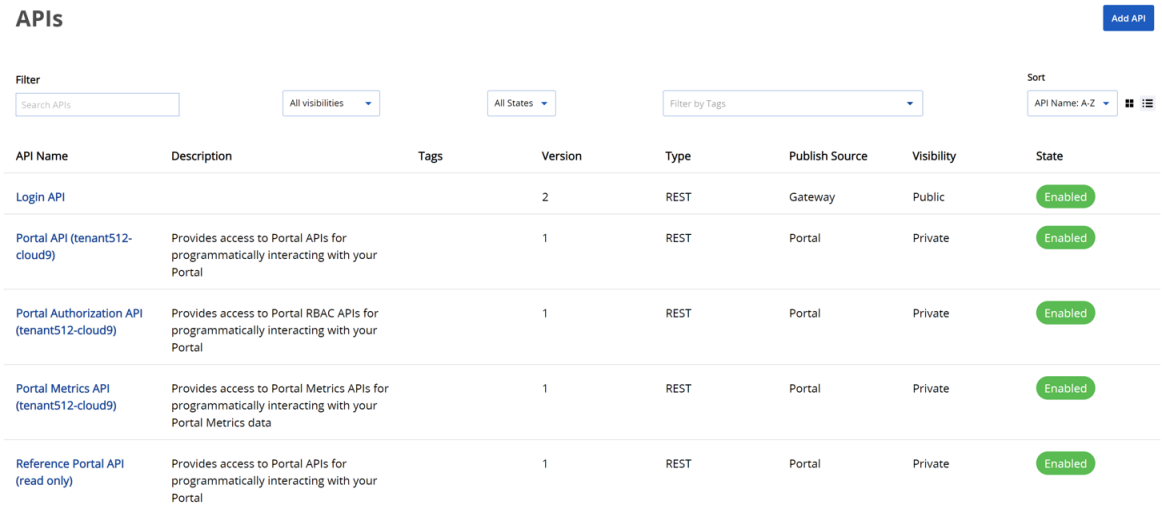
You can also manage your APIs by way of the Portal API (PAPI) or use this API in your scripts for managing APIs.

For more information about the Portal API, see Portal API (PAPI).

For more information about how to manage API visibility and permissions, see Create and Set Permissions for APIs.

Examine APIs

You can find, filter, and examine published APIs using the APIs page using either the Grid View or List View. You can also filter APIs by tags. The following image shows the APIs page:



This page shows the following information about the APIs:

- **Portal State:** The state of the API on API Management SaaS:

- **Enabled:** Org Admins and Developers can add only enabled APIs (checkmark) to applications. Applications can consume only enabled APIs.
- **Incomplete:** Incomplete APIs are APIs that have been created but do not have the required fields specified to be enabled. After you have supplied these values, you can enable the API.

  NOTE
  SOAP APIs require a Web Services Description Language (WSDL). SOAP and REST APIs require the following:

  - At least one policy template.
  - Values for the required custom fields.
- **Deprecated:** Deprecated APIs display with a (down arrow). If an API is already added to an application and the state of the API changes from **Enabled** to **Deprecated**, the application can continue to consume the API.
- **Disabled:** Disabled APIs display with an (X). If an API is already added to an application and the state of the API changes from **Enabled** to **Disabled**, the application cannot continue to consume the API. If an API is already added to an API group and the state of the API changes from enabled to disabled, the following message appears on the Edit API Group page:

  ```
  This API Group contains disabled APIs and will not be available for applications of same organizations.
  ```
- **Unpublished:** Unpublished APIs (up arrow in cloud) are APIs that have been published on the API proxy but aren't enabled in API Portal. To learn how to enable a Gateway-published API, see Enable a Gateway-Published API.

- **Visibility:** Switch to **List View** or use **Filter by** to see the visibility of APIs. Private or semi-private APIs are available only to the organizations to which they have been assigned. Public APIs are available to all organizations. Publishers can access all APIs.
- **Version:** Shows the current version of the API.
- **Tags:** Shows the tags that are associated with the API. You can add up to 25 tags to an API.
- **Applications:** The number of applications that have added the API.

  NOTE
  You can edit only some of the API settings for applications that have added APIs.

View the List of Applications that Have Added an API

Prerequisite: You have added at least one API to an application.

For more information about how to add APIs to applications, see Manage Applications

On the APIs page, in the row of the API, select the number in the **Applications** column. The Applications page opens with a list of the applications to which an Org Admin or Developer has added the API.

View the Details for an API

Prerequisite: You have created the API.

For more information about how to create APIs, see Create and Set Permissions for APIs

Follow these steps:

1. From the menu bar, select **Manage**, **APIs**.
   The **APIs** page appears.
2. Select the API for which you want to view the details.
   The page with Overview information opens.

   NOTE
   The **Spec** tab is only visible for REST APIs. The **Deployments** tab is only visible to Portal Admins and API Owners.
3. View the information in the **Overview** and **Deployments** tabs.

NOTE
You can download API assets such as WSDL or Swagger from the Overview page.

4. View your API documents on the **Documentation** tab.
   For more information about this tab, see Manage API Documents.

View How and Where an API is Deployed in the Details Page

On the APIs page, click the name of the API that you want to view deployment details. The **Deployments** tab includes a list of proxies on which the API is deployed. The APIs that are published but not deployed display the enrolled proxies with the **Not Deployed** label.

For each proxy enrolled with API Portal, the deployment state, API deployment type, and the date and timestamp displays:

- **Deployment State:** When an API is deployed, undeployed, or edited, the deployment state is updated.
  The deployment states are:
  - Deployed (**green**). The current version of the API on the API Management SaaS is deployed to the API proxy shown.
  - Pending Deployment (**yellow**). The deployment process is running. Deployment can take up to 2 minutes (6 minutes for the Hybrid solution).

    TIP
    If the deployment state of the API changes on the API proxy while the API Details page is open, you can display the new deployment state faster by refreshing the browser. If an API has been stuck in the **Pending Deployment** state for a long time, there might be an issue with the deployment to the specified proxy. For on-demand and scripted API deployment types, select the deployment state link for a more detailed deployment response message.
  - Error (**red**). If the API deployment state is in **Error** state, there is a deployment or connection problem with the specified proxy.

    TIP
    To verify the state of a proxy, access the API Proxy Details page for each proxy. Then, check whether the APIs section is displaying a red **x** icon. If so, there is a problem reaching the API proxy. Contact your system administrator in this case. The red **x** icon on Proxy Details page does not apply to on-demand and scripted deployments.

    For on-demand and scripted API deployment types, select the red **Error** icon on the API Details page to view more details about the error.
- **API Deployment Type:** The API deployment type determines how Portal published APIs are deployed to the proxy. The API deployment type cannot be edited once APIs are deployed to the proxy. The API deployment type is selected when adding an API proxy. A proxy supports the following API deployment types:
  - Automatic
    Any changes to APIs are automatically deployed to the proxy.
  - On-Demand
    API deployments are triggered on-demand by calling the deployment APIs. You can access these APIs from the APIs page.
  - Scripted
    API deployments are integrated into your existing CI/CD workflow by using the deployment APIs and invoking them from your deployment script.

  For more information about deployment types, see Deployment Types.
- **Last Updated timestamp:** The details page also displays the date and timestamp of the last deployment attempt. The timestamp is updated on successful and failed deployments.

Using Search

See the following notes about search:

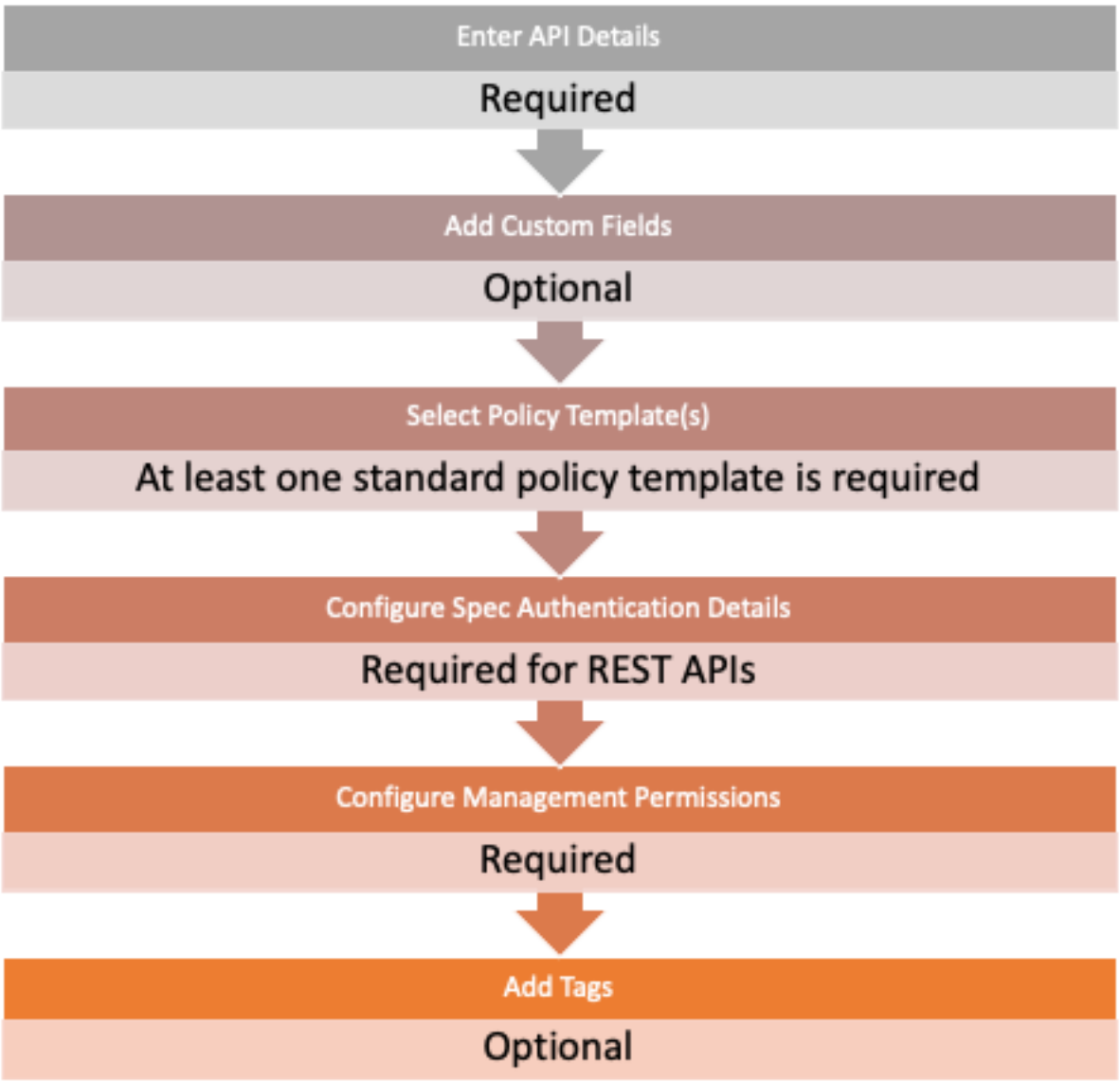
- Helps you find APIs and content on custom pages (pages that the Portal Admin publishes using the content management system).
- Finds exact matches, and partial matches in titles, names, descriptions, and other content.
- Does not find content on standard pages (for example, the Applications and Analytics pages).
- Does not support Boolean searches (operators like AND, NOT and OR to get more relevant results).

Create and Set Permissions for APIs

This article includes information about how to add APIs and set up API management permissions and visibility.

> **NOTE**
>
> **Configuration of Organization Access and Visibility Permissions Moved for API Portal Version 5.1.2**
>
> For the release of version 5.1.2, these settings are now configured directly in the API Details for each individual API.

The following diagram summarizes the workflow for adding and setting up your API using the Add/Edit API wizard:



Configure API Details, Custom Fields, Policy Templates, Spec Authentication and API Management Permissions

Follow these steps:

1. Log in to API Management SaaS as a Portal Admin.
2. From the menu bar, select **Manage**, **APIs**.
   A list of APIs appears.
3. Do one of the following steps:

– To set up API details and policies to a new API, select **Add API**.
– To set up API details and policies to an existing API, click the API for which you want to edit details. On the *API Details* page, select **Actions, Edit API Details**.

The **Add/Edit API wizard** opens.

4. In the **Details** section, choose between **REST** and **SOAP** API types. If applicable, select **Choose file** to upload your Swagger or Web Application Description Language (WADL) definition files (for REST API), or Web Services Description Language (WSDL) file and optional XSD file (for SOAP API). For more information, see About API Description Files. If you do not have definition files, provide API details manually.

   If you uploaded an API definition file, the fields are already filled with values. You are alerted to any mandatory fields that do not have assigned values.

   Provide values as follows, and then select **Save & Next**:

Field	Notes
API Name	Maximum name length is 255 characters. Name must be unique.
Publish State	Choose one of the following options: • **Enable:** The API can be added to applications and applications can consume them. The API will be published. • **Disable:** The API cannot be added to applications. If an API is already added to an application and the state changes from enabled to disabled, the application cannot continue to consume the API. • **Deprecate:** The API cannot be added to applications and deployed to proxies.
Version	The value for this field can only contain 0-9 and be delimited with . _ and - characters.
Location of API	The API proxy routes requests from applications to the location of the API behind the API proxy. Developers do not see this information. Use a context variable to Route the API to Multiple Data Centers.
Access	Choose one of the following options: • **Public:** The API is visible to all organizations. If you set the visibility to Public, any API Owner added in the future will automatically see this API. • **Private:** The API is visible only to you. If you are part of an organization, the organization can see the API as well. API Owners can still see this API regardless of the settings. Set up which organizations can see this API. If you are not part of an organization and creating a private API, then only API owners will be able to see it. APIs that you set the visibility to **Private** are not associated with an organization.
API EULA	Select an available End User License Agreement (EULA) to assign to this API. You can assign EULAs to APIs that are not already associated to an organization. Before Developers can get an API key for the API, they must agree to your EULA.

Field	Notes
Public Description	The description appears in the API Explorer and in the Add/Edit Application wizards. Provide Developers with API information, such as its proxy URL and authentication requirements. **Required:**No Maximum description length is 255 characters.
Private Description	Maximum description length is 255 characters.
API URI	Provide the **API Proxy URL**, which is the public URI of the API on the API Proxy. This URI is part of the URL used by developers in their web/mobile applications to send requests to the API.

5. If you have enabled Custom Fields, the **Custom Fields** section opens. Custom fields are defined by your Administrator to provide additional metadata for your API. Complete the custom fields values, and then click **Save & Next**.
   The **Policy Templates** section opens.
6. Do the following, and then select **Save & Next**:
   a. Select your desired policy templates from the drop-down menu.
   b. Expand on an added policy to set its parameters.
   c. Combine multiple policy templates. Ensure that you select them in the order that you want the API proxy to apply them.
   For more information about how to control API access with policy templates, see Policy Templates.

   NOTE
   To learn how to implement a rate limit and quota with a policy template, see this page.

   (REST APIs only) The **Spec Authentication** section opens.
7. Select the **Authentication type** and provide authentication details as needed. For REST APIs, the Spec Authentication tab will allow you to configure the authentication settings to test your API via the Spec tab. Note that the Spec Authentication tab and testing via the Spec tab is not available for SOAP APIs. The selected authentication type is used in the **Spec** tab of the details page when trying out the API. Click **Save & Next**.

Set Up API Management Permissions

Set up who has the permissions to edit and delete this API.

Follow these steps:

1. After setting up the API details, policy templates, and spec authentication in the Add/Edit API wizard, the **Management Permissions** section opens. Alternatively, you can navigate to an existing API, and then click **Actions > Edit Management Permissions**.
2. Select who can manage the API. You must specify a managing organization or at least one API Owner:
   a. **Select the Managing Organization:** Selecting a managing organization allows all users within that organization to edit this API. This only applies if the user has permissions to edit APIs.
   Only Publisher organization type is displayed here as a Consumer organization type cannot be a managing organization.

   NOTE

   Retain Visibility?

   If you change the managing organization of an API to a different organization, retain the visibility of the previous managing organization by clicking **Yes** in the **Retain Visibility** pop-up dialog if you intend to allow existing applications of the previous organization to be consumed.

   b. **Select API Owner Permissions**

- **Open:** Specify that anyone with API management permissions can edit this API.
- **Restricted:** Specify users with API management permissions to edit this API.

  NOTE
  If a Portal Admin or an API Owner added an API in API Management SaaS version 4.4 and assigned an Org User to the user permission list:

  - After the upgrade to API Management SaaS version 4.5, only the Portal Admin or the API Owner who belongs to the permissions list has access to manage this API.
  - An Org User who was a part of the permissions list will gain or lose the API management permissions based on the number of organizations associated with this API. If there were multiple organizations assigned, the Org User will lose the API management permissions.

3. Select **Restricted**.
   A list of API Owners appears.
4. Select the users that have permission to edit and delete the API.
   The selected users appear in the right column **Selected**.
5. Click **Save & Next**.

The API management permissions are updated. The selected users can edit and delete the API.

Set up Organization Access

For private APIs, the Organizations Access section lets you assign access or visibility rights to individual organizations for the API and assign rate limits and quotas at the 'API per organization' assignment level.

For public APIs, you may assign rate limits and quotas at the 'API per organization' assignment level only in this section of the form.

Turning Visibility Permissions On or Off (For Private APIs only)

To give an organization access to the private API, select any of the listed organizations and click **Visibility On**. Organizations with that are given access to the API will have the access toggle 'on' or colored in blue.

Similarly, if you need to remove access to the API from an organization, select the organization and click **Visibility Off**. Organizations that do not have access will have the access toggle 'off' or colored in gray.

Adding a Rate Limit and Quota

You may add a rate limit and quota with an 'API per organization' assignment level to the API here. Follow these steps:

1. Select an organization, and click **Add Rate Limit and Quota**.
2. In the Assign Rate Limit and Quota window, select a rate limit and quota from the drop-down list.
3. Click **Save**.

   TIP
   If you have a long list of organizations to navigate, use the Filter box to search for your organization by keyword, or use the **All**, **Access**, or **No Access** preset filter buttons.

Add Tags

Adding tags is a way to group categories of relevant APIs. Tags appear on the APIs' list view and card view. API consumers can then search and discover APIs according to their visibility permissions using these tags.

To add one or more tags, select the required tag(s) from the Available Tags list. You can filter the tags by entering keywords in the search box or by clicking Select swagger tags to auto-select all swagger tags from the list.

You can add up to 25 tags to an API. Maximum tag length is 60 characters.

Only global and API level tags are added. Endpoint level tags are not be shown.

After completing all the required sections of the Add/Edit API wizard, your new API shall be created with the status that was assigned in the Details section.

View the APIs and Applications on the Developer Console

You can view the applications and APIs that the API Owner or Portal Admin creates on the Developer Console.

In this article:

View All APIs

Follow these steps:

1. Log in to API Portal as an Org Admin or Developer.
2. From the menu bar, select **Manage**, **APIs**.
   The **APIs** page opens showing all the APIs. This page displays a list of all the APIs along with each API's state (Available or Unavailable). APIs in the "Available" API state have been deployed on the Gateway. APIs in the "Unavailable" API state are pending deployment on the Gateway.
3. To view the Swagger UI for the API, click **Spec**.
   – If the API includes a Swagger file, a parsed Swagger file is displayed on the right side of the page. In the **Resources** section, you can select a link that jumps to the associated Argument in the middle of the page and to the associated definition in the parsed Swagger file.

     NOTE
     If you select a Web Application Description Language (WADL) or Swagger file in the API Definition and you want to expose your Proxy URL to Developers, ensure that the file includes the Proxy URL information.
   – If the API does not include a Swagger file, the right side of the page is blank. In the **Resources** section, each Proxy URL for the API is listed. In the middle of the page, each Proxy URL and its API deployment status is displayed. You can copy the Proxy URL from this section and use it as required.
4. If the API includes a Swagger file, to download the Swagger file, click **Swagger File**.
5. To add the API to an application:
   a. Click **Use API**.
      A drop-down list of applications displays.
   b. Select the application to which you want to add the API, and then click **Accept**.
      A check mark beside the application indicates that the API is already added to the application. The EULA is displayed.
      The App details page opens showing the APIs that have been added to the application.

View All Applications

Follow these steps:

1. While logged in to API Portal as an Org Admin or Developer, from the menu bar, select **Manage**, **Applications**.
   The **Applications** page opens showing all the APIs. This page displays a list of all the applications along with each application's state (Available or Unavailable). Applications in the "Available" state have been deployed on the Gateway. Applications in the "Unavailable" state are pending deployment on the Gateway.
2. Select an application.

The application page opens to the **Configuration** tab, where you can view a list of the API keys that have been added to the application. You can view the following application details:

Configuration Field	Description
API Key (Client ID)	Uniquely identifies the consuming the API. **Read-only:** Yes
Shared Secret	Displays the credentials for API authentication. **Read-only:** Yes
Status	The application's state. Applications in the "Enabled" state indicate that you can get an API key and that you can see it in API Explorer. Applications in the "Disabled" state indicate that you do not have permissions to interact with the APIs that have been added to the application. **Read-only:** Yes
OAuth Callback URL(s)	The authentication mechanism for consuming the API. **Read-only:** Yes
OAuth Scope	The authentication mechanism for consuming the API. **Read-only:** Yes
OAuth Type	The authentication mechanism for consuming the API. **Read-only:** Yes

3. To view the APIs that have been added to the application, click the **APIs** tab.
4. To add more APIs to the applications, select **Actions**, **Edit API Assignment**.
5. To view and edit the application, select **Actions**, **Edit Application**.
   The **Details** page opens and displays the fields described in the following table.

Notes about using Search

Search for APIs by entering a partial or full name of the API you are looking for.

Use search:

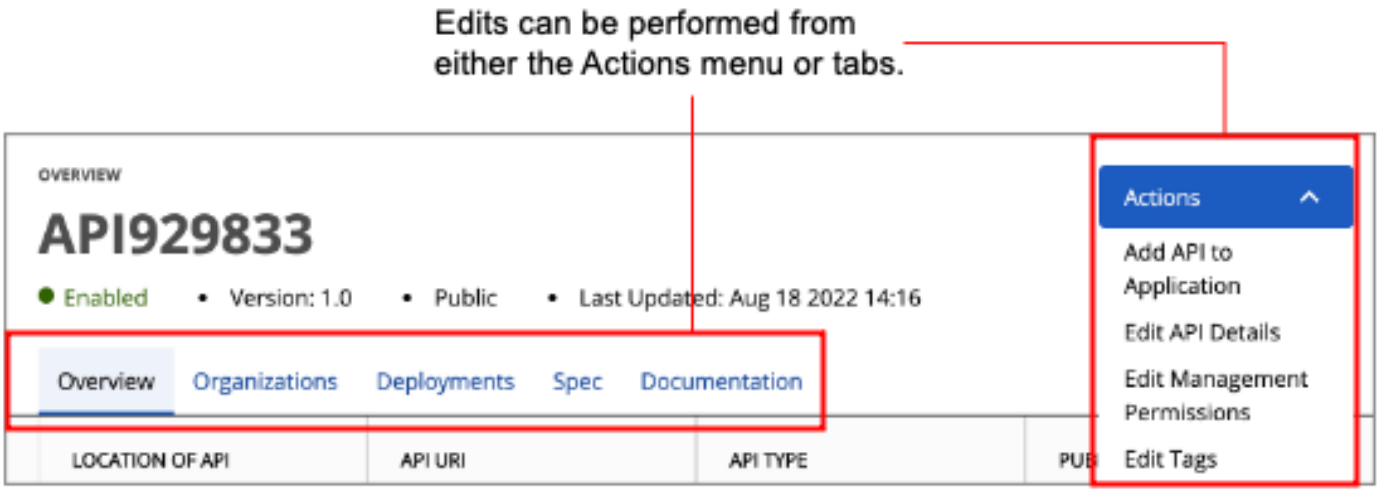
- To help you find APIs and content on custom pages (pages that your Admin publishes using the content management system).
- To find exact matches, and partial matches in titles, names, descriptions, and other content.
- Search does not find content on standard pages (for example, Applications and Analytics pages).
- Search does not support Boolean searches (operators like AND, NOT and OR to get more relevant results).

Edit and Delete APIs

Publishers can add, edit, and delete APIs. You cannot edit some settings if the API is in use or if the API is a Gateway-published API. For more information about how to add APIs (including the configuration of policy templates, management permissions, and tags) see Create and Set Permissions for APIs.

Editing an API may involve

- Performing an 'action' to change core settings (e.g., API details, policy templates, etc) that were made when the API was created or last revised.
- Selecting a tab in the API Details page for additional configuration options for the API.



	Portal Admin	API Owner	Org Publisher	Org Admin	Developer	Guest	Anonymous
Add API	✓	✓	✓	✗	✗	✗	✗
Read API	✓	✓	✓	✓	✓	✗	✗
Update API	✓	✓	✓	✗	✗	✗	✗
Delete API	✓	✓	✓	✗	✗	✗	✗
Assign Managing Org	✓	✓	✗	✗	✗	✗	✗
Assign User Permissions	✓	✓	✗	✗	✗	✗	✗
Assign Consuming Org	✓	✓	✓	✗	✗	✗	✗

NOTE
The Org Publisher has publishing permissions only in their assigned organization.

In this article:

Edit an API: Performing Actions

To perform an action on an API:

1. Log in to API Management SaaS as a Portal Admin.

2. From the menu bar, select **Manage**, **APIs**.
   The APIs are listed on the APIs page.
3. Select the API that you want to edit.
4. Select **Actions**, and then select one of the following options based on the action that you want to perform:
   – Add API to Application
   – Edit API Details
   – Edit Visibility Permissions (Removed as of API Portal Version 5.1.2)

     NOTE
     The management of private API visibility for organizations has been moved to the Organizations tab on the API Details page.
   – Edit Management Permissions
   – Edit Tags
   – Delete API (action unavailable if the API is published and used in an application)

     NOTE

     - To edit policy templates and/or spec authentication, select the Edit API Details action first.
     - For more information about how to edit API details, management permissions, and tags, see Create and Set Permissions for APIs

     .

Edit an API: Using Tabs

1. Log in to API Management SaaS as a Portal Admin.
2. From the menu bar, select **Manage**, **APIs**.
   The APIs are listed on the APIs page.
3. Select the API that you want to edit.
   The API Details page with information about the selected API opens. By default, the **Overview** tab opens first, displaying details such as

   - API location, URI, and type
   - Publish source (e.g., Portal or Gateway)
   - Public/private description
   - Usage (e.g., applications)
   - Assets and tags associated with the API
4. Select the tab associated with the configurations that you want to view and/or edit:
   – **Organizations**: Lists all organizations that currently have access or don't have access to the API. The currently assigned rate limit and quota is also displayed for each organization at the API per organization level (each organization can be assigned a custom API per organization rate limit and quota OR the default one). From this tab, you may also manage private API access for organizations and assign rate limit and quotas at the API per organization assignment level.
   – **Deployments:** View the proxies to which the API is deployed and the type of deployment (On Demand, Automatic, or Scripted).

     NOTE
     This tab is visible to Portal Admins and to users who have **read** permissions for APIs, proxies, and API deployments.

     For more information about how to use this tab, see Manage API Deployments.
   – **Spec:** View the Swagger API documentation and test and explore your API.
     For more information about how to use this tab, see Test and Explore APIs.
   – **Documentation:** View the markdown content for the API.

NOTE
This tab is visible to Portal Admins.

For more information about how to use this tab, see Manage API Documents.

Add an API to an Application

You add APIs to applications while adding or editing an application.

1. From the Actions menu, select **Add API to Application**
   The Add API to Application pop-up window appears. Applications eligible for the addition of the API are shown. Applications with a checkmark indicate that they already contain the API.
2. Select the application in the list to add the API to.

For more information about how to add available APIs to an application, see Manage Applications.

Manage Private API Access for Organizations

Only private APIs can have their access (formerly 'visibility') managed for select organizations in the **Organizations** tab of the API Details page.

To give an organization access to the private API, select any of the listed organizations and click **Add Access**. Organizations that are given access to the API will have the access toggle 'on' with a check mark.

Similarly, if you need to remove access to the API from an organization, select the organization and click **Remove Access**. Organizations that do not have access will have the access toggle 'off'.

Manage Rate Limit and Quotas: API Per Organization

To add or edit a rate limit and quota with the API per organization assignment level:

1. In the **Organization** tab of the API Details page, select an organization, and click **Add Rate Limit and Quota**.
2. In the Assign Rate Limit and Quota window, perform one of the following actions:
   – Select a rate limit and quota from the drop-down list and click **Save**.
   – Click **Restore Default** to assign the default value of the system API per organization rate limit and quota.

See also: Manage Rate Limits and Quotas.

Delete an API

Portal Admins and API owners can delete APIs. Deleting an API Management SaaS-published API from API Management SaaS also deletes the API on the API proxy when API Management SaaS and the proxy synchronize.

NOTE
Deleting *Gateway-published* APIs on API Management SaaS does not automatically delete the API on the API proxy. Therefore, the API Proxy page shows a synchronization error. Have an API proxy administrator remove the `Set as Portal Managed` assertion from the API.

Follow these steps:

1. From the APIs page, disable the API:
   a. Select the API that you want to disable.
      The API Details page opens.
   b. Select **Actions**, and then select **Edit API Details**.
      The **Details** tab opens.
   c. Select **Disable** as the publish state, and then select **Save**.

2. Determine which applications include the API that you want to delete, and remove the API from each application that includes it:
   a. From the API Details page (with the **Overview** tab open), select **Applications**.
      The Applications page opens and shows a list of all applications that include the API.
   b. From the drop-down for the application from which you want to remove the API, select **Edit**.
      The **Details** page in the Application wizard opens.
   c. Select the **API Management** tab.
   d. In the **Selected APIs** section, select the **x** next to the API that you want to remove from the application, and then select **Save**.

      NOTE
      If an application includes only one selected API, you can remove the API only by replacing it with another API.
   e. Verify that there are no applications that include the API that are pending approval. If there are applications pending approval, reject them.
3. Remove the API from each API group that includes it:
   a. From the API Details page (with the **Overview** tab open), select **API Groups**.
      The API Groups page opens and shows a list of all the API groups that use the API.
   b. From the drop-down for the API group from which you want to remove the API, select **Edit**.
      The Edit API Group page opens.
   c. In the **APIs** section, under **Selected APIs**, select **Remove** next to the API that you want to remove from the API group, and then select **Save**.
4. Delete the API. On the API Details page, select the **Actions** menu, and then select **Delete API**.
5. (Recommended) If the API is a Gateway-published API, have an API proxy administrator complete the following:
   a. In the Policy Manager, to log in to the API proxy as an administrator.
   b. Remove the `Set as Portal Managed` assertion from the API on the proxy.

The API is deleted from API Management SaaS.

Manage API Tags

Add Tags to an API

Adding tags is a way to group categories of relevant APIs. Tags surface on the **APIs** page. Consumers can then search and discover APIs according to their visibility permissions using these tags.

In the **Tags** section of the Add/Edit API wizard, the tags that are imported from the Swagger document and those available in API Management SaaS are listed.

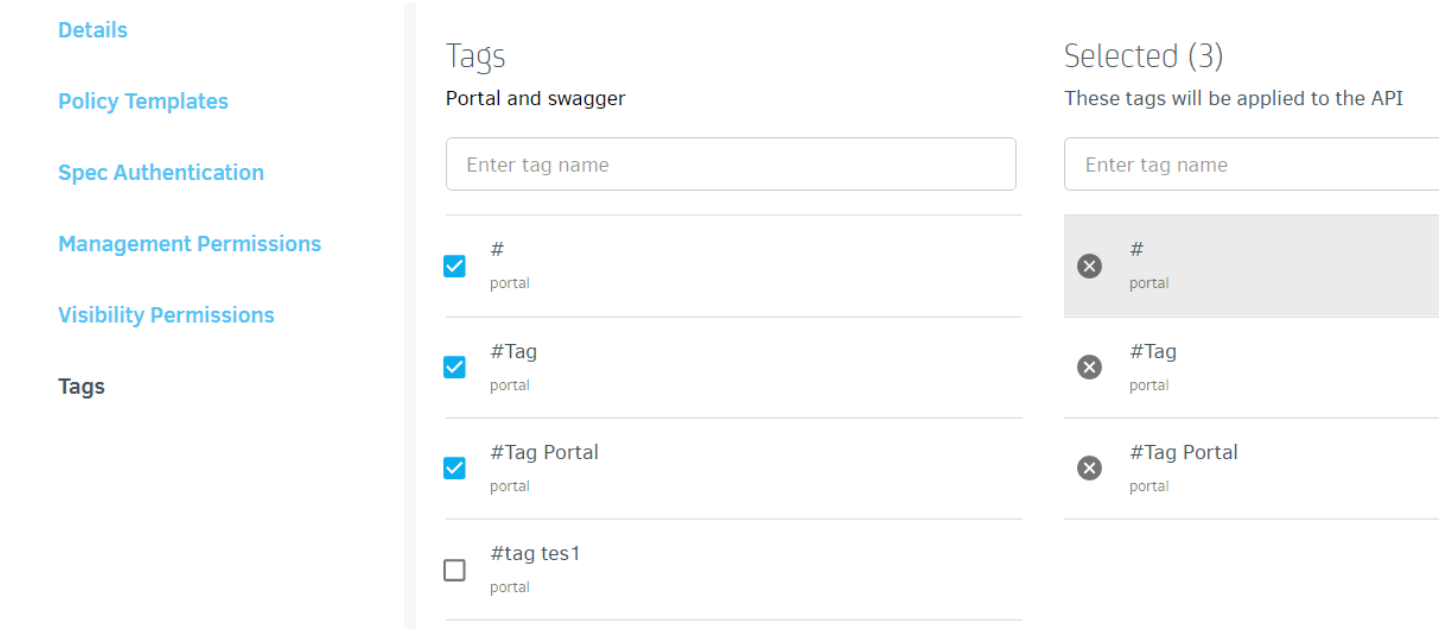
You can add up to 25 tags to an API. Maximum tag length is 60 characters.

NOTE
All restrictions on the tag naming convention are removed and all characters are now allowed in tag names.

Follow these steps:

1. In the Add/Edit API wizard, you can add tags in the **Tags** section. Alternatively, you can navigate to an existing API and click **Actions**, **Edit Tags**.
2. Select the tags that you need to associate with the API.
   The tags are added to the **Selected** column.

   TIP
   Click **Select swagger tags** to select all the tags that have been imported from Swagger with a single click.



3. To add a tag, enter the tag in the **Enter tag name** field, and then click **Add New Tag**. The tag is added and it is selected to be applied to the API.
4. Click **Save & Next**.

The selected tags are added to the API.

> **TIP**
>
> Portal Administrators can filter tags using the **inUse=True/False** filter with the new endpoint in the Portal API (PAPI):
>
> `GET /api-management/1.0/tags?inUse=False` to get all API tags that are not associated with any API.

Publish APIs with Additional Configurations with the API Portal

Portal-published APIs typically have simple policies that are based on one or more policy templates. This page describes how to add Portal-published APIs with additional configurations to augment their operational behaviour.

> **NOTE**
> Use only API Management SaaS-published APIs in a multi-cluster environment where your API Management SaaS is integrated with multiple API proxies.
>
> If you fail to migrate a Gateway-published API in a multi-cluster environment, the API might appear and then disappear from the APIs section in API Management SaaS. To resolve this issue, use the Gateway Migration Utility (GMU) to migrate all Gateway-published APIs to all clusters. To learn more about the GMU, see "Gateway Migration" in the API Gateway online documentation.

For more information about Gateway-published APIs, see Publish APIs with the API Proxy and Policy Manager.

In this article:

Route an API to Multiple Data Centers

To route an API to multiple data centers, use context variables in the API location. For example, enter the API location:

http://localhost:8080/echo?name=${gateway.cluster.hostname}

The value in the context variable *${gateway.cluster.hostname}* is used for routing in policy for the API.

For more information about context variables, see "Context Variables" in the API Gateway online documentation.

Add an API Specification File to the API

Upload a well-crafted definition (specification) file for each API Management SaaS-published API. This can be a WADL .xml or Swagger .json for REST APIs. For SOAP APIs, the attached WSDLs and XSDs can be downloaded for use in any external SOAP client like SOAP UI.

Without a definition file, you cannot test the API and Developers cannot use the API Explorer or Swagger UI to try the API using the API Explorer or Swagger UI. For more information, see the following specifications:

- WADL
- Swagger
- WSDL
- XSD

> **NOTE**
> API Explorer is only accessible through the API Management SaaS/Ingress tenant. If you are using an external tenant, test and explore APIs using the Swagger UI instead. For more information about testing with Swagger UI, see Test and Explore APIs.

The API Explorer consumes only APIs that have secure (HTTPS) endpoints. If an API has a secure and unsecure endpoint, ensure that your definition files point to the secure endpoint.

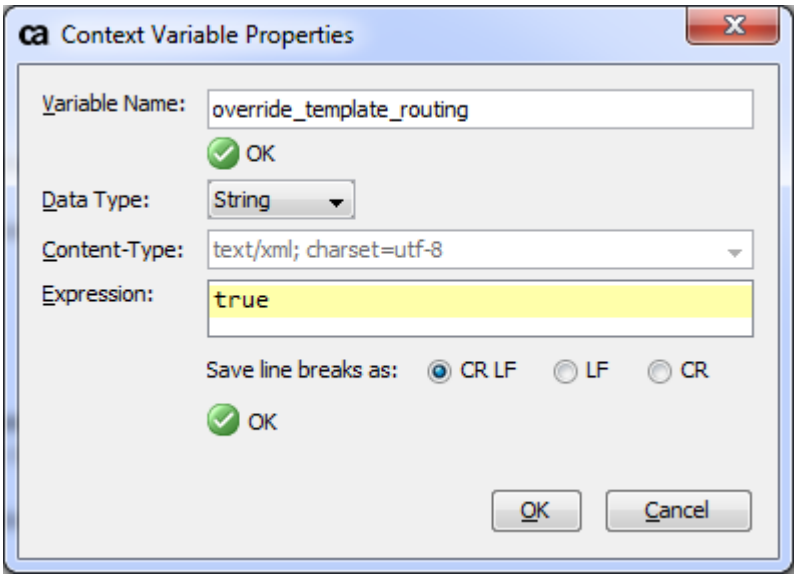
> **NOTE**
> If you are uploading a definition file and you want to expose your Proxy URL to developers, make sure the file contains the Proxy URL information.

Customize Routes

When API Owners add a API Management SaaS-published API, they specify the location of the API. API proxy administrators can use the Policy Manager to customize how the API proxy routes calls to the API.

Follow these steps:

1. Use the Policy Manager to log in to the API proxy as an administrator.
2. Open the API policy in the Policy Development window.
3. Locate and double-click this line in the policy:
   `Set Context Variable override_template_routing as String to: false`
4. In the **Context Variable Properties**, change **false** to **true**.
5. Select **OK**, and then save and activate the policy.

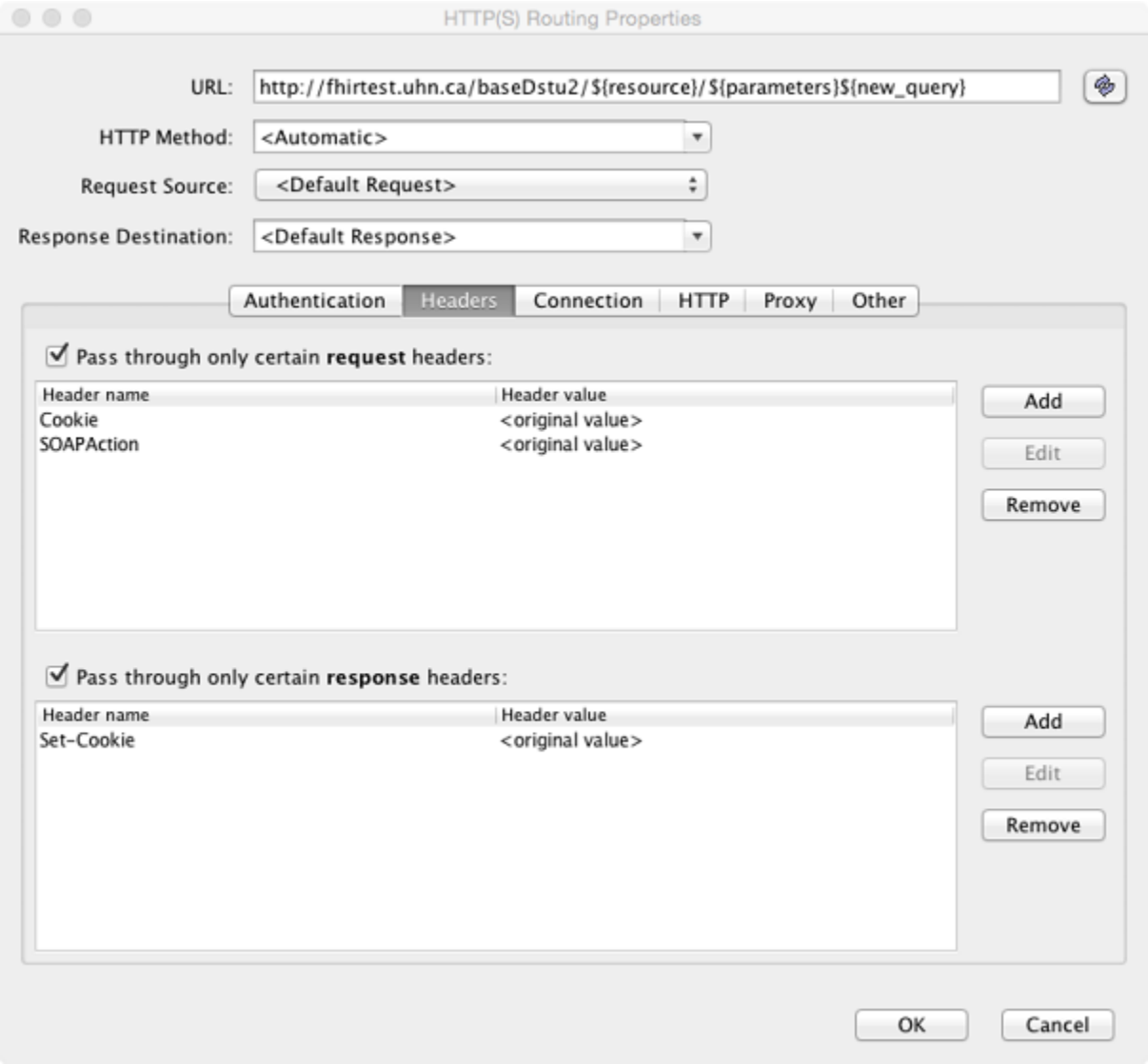


If route customization is enabled on an API, when someone uses the API Explorer to send a request to the API, the response contains an erroneous Access-Control-Allow-Origin header value that the API Explorer cannot process. To prevent this problem, the API proxy administrator must remove the Access-Control-Allow-Origin CORS header from the custom routing section of the API policy. One way to remove that header is to configure the HTTP routing properties to pass only certain response headers.

Filter Response Headers

Follow these steps:

1. In the Policy Manager, open the API Route using HTTP(S) assertion. The HTTP(S) Routing Properties dialog opens.
2. On the **Headers** tab, select the checkboxes **Pass through only certain request headers** and **Pass through only certain response headers**. Do not change the headers.
3. Select **OK**, and then save and activate the policy.



For more information about routing, see "Route via HTTP(S) Assertion" in the API Gateway online documentation.

Control API Access with Policy Templates

You can use policy templates to customize how a policy on the API proxy processes calls to an API. API publishers commonly use authentication and quota policies to control API access.

> **NOTE**
> When you publish an API, you can include multiple policy templates. If you select multiple policy templates, ensure that you select them in the order that you want the API Proxy to apply them. Also note that some templates might be incompatible with other templates. Conversely, some templates might need to be combined with another template.
>
> For more information about your policy templates, contact your API proxy administrator.

Out of the box, the API wizard also provides sample policy templates, which API proxy administrators can revise. The default policy templates provide samples for authenticating calls to APIs, and for managing API usage.

Authentication policy templates provide different options for authenticating calls to APIs:

- No authentication
- API Key authentication
- OAuth 2.0

The available policy templates are as follows:

Policy Template	Template Type	Recommended Environment	Notes
API Key	Standard	Testing only	Applies an API Key check to all API access calls.
No Auth	Standard	Testing only	Applies no authentication check. Useful for proxying third-party APIs (such as Twitter) that have their own authentication requirements.
OAuth 2.0	Standard	Production	Applies an OAuth 2.0 check to all API access calls. Appropriate for both two and three-legged OAuth implementations. Supports the following grant types: • Implicit • Client credentials • Resource owner password credentials • Authorization cod
l7.apim.system - Rate & Quota Policy Template - 2.0 (Available as of Portal 5.1.1 SaaS)	API management		Restricts the number of times that all applications can query an API in a second AND/OR defines the number of times that an API can be queried in a defined interval. Version 2.0 effectively replaces version 1.0 (see next row) with the addition of the 'API per Organization' assignment level.
l7.apim.system - Rate & Quota Policy Template - 1.0	API management		Restricts the number of times that all applications can query an API in a second AND/OR defines the number of times that an API can be queried in a defined interval. This policy effectively replaces the legacy Rate Limit, Quota by Month and Quota by Day policies.
Rate Limit Policy (Deprecated as of Portal 5.1)	API management		Restricts the number of times that an API can be queried in a second. For example, a rate limit of 1 prevents all the applications that use that API from accessing it more than once per second.

Quota By Day Policy (Deprecated as of Portal 5.1)	API management		Restricts the number of times that an API can be queried in a day. For example, a quota limit of 1 ensures that all the applications that use that API can only access it once per day.
Quota By Month Policy (Deprecated as of Portal 5.1)	API management		Restricts the number of times that an API can be queried in a month. For example, a quota limit of 1 ensures all the applications that use that API can only access it once per month.

The standard policy templates have the same parameters:

- Debug mode: When troubleshooting the API, turn on Debug mode to get verbose responses.
- Email: The address used by the SMTP server to send an email alert.
- SLA: The Service Level Agreement (SLA) period expressed in milliseconds. If the API does not reply within the SLA period, the SMTP server sends an email alert to the email alert address.
- SMTP Server: Your email gateway.
- SSL: To secure API calls between the API proxy and applications, select SSL.

Configure Rate Limit and Quota During API Publishing

NOTICE
As of Portal 5.1, users are advised to use the streamlined Rate & Quota policy template to regulate API usage when creating and editing APIs. Legacy templates such as Rate Limit, Quota by Month, and Quota by Day have been deprecated as their functions have been consolidated under the Rate & Quota policy template.

Prerequisites:

- The Portal admin has deployed the Rate & Quota Policy template system bundle to the API proxies to which you want the API deployed. See Manage Policy with Gateway Bundles to learn more.
- Rate limits and quotas have already been defined in the API Portal and you understand which rate limits and quotas to apply to the API. Recall that there are multiple assignment levels that impact whether the constraints are imposed on the organization or the API itself.
- You must be logged into API Portal as a publisher, or as an Org Admin or Developer with API publishing capabilities.

Organizations can manage API usage using one of the following policy templates:

- **Rate & Quota**
  Restricts the number of times that all applications can query an API in a second AND/OR defines the number of times that an API can be queried in a defined interval. This policy effectively replaces the legacy Rate Limit, Quota by Month and Quota by Day policies (see other policies described in this list).
- **Rate Limit (Deprecated as of Portal 5.1)**
  Restricts the number of times that an application can query an API in a second. For example, a rate limit of 1 prevents all the applications that use that API from accessing it more than once per second.
- **Quota By Month (Deprecated as of Portal 5.1)**
  Restricts the number of times that an application can query an API in a month. For example, a quota limit of 1 ensures that all the applications that use that API can only access it once per month.
- **Quota By Day (Deprecated as of Portal 5.1)**
  Restricts the number of times that an application can query an API in a day. For example, a quota limit of 1 ensures that the applications that use that API can only access it once per day.

To assign a rate limit and quota with an 'API' assignment level:

1. Add an API or edit an existing API.
   For more information about how to add or edit an API, see Edit and Delete APIs or Create and Set Permissions for APIs.
2. In the **Policy Templates** section of the API wizard, select **l7.apim.system - Rate & Quota Policy Template - 2.0**.
3. Select a preconfigured rate limit and quota with an API assignment level to define the access limits for this API and then click **Save & Next**.

To assign a rate limit and quota with an 'Organization' level:

1. Add an organization or edit an existing organization.
2. In the Rate Limit and Quota section of the Overview tab, select a preconfigured rate limit and quota with an organization assignment level to define access limits for the organization and click **Save**.

To assign a rate limit and quota with an 'API per Organization' assignment level:

1. In the **Organization** tab of the API Details page, select an organization, and click **Add Rate Limit and Quota**.
2. In the Assign Rate Limit and Quota window, perform one of the following actions:
   – Select a rate limit and quota from the drop-down list and click **Save**.
   – Click **Restore Default** to assign the default value of the system API per organization rate limit and quota.

Publish APIs with the API Proxy and Policy Manager

This page describes how to create Gateway-published APIs. Gateway-published APIs use sophisticated policies. You can only add policy templates to Gateway-published APIs using Gateway. The API proxy administrator publishes APIs using the Policy Manager and the API proxy.

In API Management SaaS, the initial state of newly created Gateway-published APIs is unpublished. The Portal Admin or API Owner then enables the API on API Management SaaS to make it available to developers. Enabling an API requires assigning the API an API EULA. Developers cannot access the APIs until the API is enabled.

> **NOTE**
>
> If you are using Gateway-published APIs in a multi-cluster environment where your Layer7 API Developer Portal is integrated with multiple API proxies, ensure that the same set of APIs is defined on all proxies. For example, do not use Gateway-published APIs in such an environment if you do not plan on defining
>
> - The same set of services/APIs on all clusters.
> - The same API name and API URL on all proxies.
>
> If the API definitions themselves differ by name/URL or different proxies have different sets of APIs then these discrepancies are carried through to the Portal; if this occurs, API details become inconsistent and APIs appear and disappear from the APIs section of your Portal. To resolve this issue, use the Gateway Migration Utility (GMU) to migrate all Gateway-published APIs to all clusters. For more information about the GMU, see "Gateway Migration" in the API Gateway online documentation.
>
> Gateway administrators should refrain from changing the name of the API on the Gateway/Policy Manager after enablement on the API Management SaaS as this name change will not transfer over to the Portal.

In this article:

Best Practices for API Proxy Administrators

- Do not copy and paste Gateway-published APIs (that is, services with the `Set as Portal Managed Service` assertion in their policy) in the Policy Manager. The reason is that the original and copy will have the same API ID.
- To allow the API Explorer to consume APIs that have policies or policy templates containing the Return Template Response to Requestor Assertion, clear the assertion's **Send Response Immediately** checkbox.
- For hybrid customers, if your on-premise Gateway requires a proxy setting for any outbound traffic or connections, modify the Routing Assertions in your specific policies or services.

For API Management SaaS-published APIs, see Publish APIs with the API Portal.

Create a Gateway-Published API

Follow these steps:

1. In Policy Manager, log in to the API proxy.
2. Add an API service to the API proxy.
   For more information about how to add an API service to the API proxy, see the API Gateway online documentation.
3. Open the API service in the Policy Manager.
4. Enable the **Set as Portal Managed Service** assertion in your API service. You can find this setting in **Internal Assertions > Set as Portal Managed Service**.
5. Use the **Include Policy Fragment** assertion to add the **Portal Service Preface** fragment to the beginning of the API service.
6. Drag the **Set as Portal Managed Service** fragment to the API service below the `Portal Service Preface` fragment.
7. Select **Save and Activate**. After the API proxy and API Management SaaS synchronize, the API appears in API Management SaaS on the APIs page.
8. Access the Service Properties for the API service. Select the HTTP/FTP tab and then enable the **Options** HTTP method.

Enable a Gateway-Published API

Each API listed on the APIs page has one of the following API Management SaaS states:

Portal State	Icon	Notes
Enabled		The API is published and available.
Unpublished		Initial state of an API. Requires acceptance of the API EULA.
Disabled		Existing applications will no longer be able to access this API.
Deprecated		You can deprecate enabled APIs. The API is still available, but use of a replacement API is preferred.

IMPORTANT
Prequisite for Gateway-published APIs: An Automatic API deployment type must be used for the API proxy

Follow these steps:

1. Log in to API Management SaaS as a Portal Admin or API Owner.
2. From the menu bar, select **Manage**, **APIs**.
   The APIs page appears.
3. Filter the list by State, selecting `Unpublished` .
4. Click the name of the API from the filtered list.
   The API Details page for that API appears.
5. Select **Actions**, **Edit API Details**.
   The Details page appears.
6. Select an **API EULA**, and then select **Save**.

The Gateway-published API is enabled.

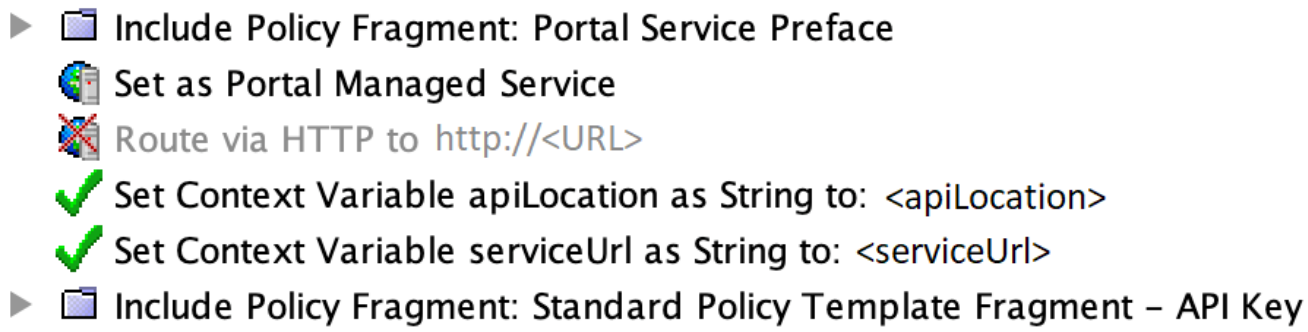
Enable Analytics for a Gateway-Published API

The following policy fragments provide analytics:

- Standard Policy Template Fragment - API Key
- Standard Policy Template Fragment - OAuth 2
- Standard Policy Template Fragment - NoAuth

Follow these steps:

1. In the Policy Manager, log in to the API proxy.
2. Open the API Service.
3. After the Portal Service Preface section, drag the **Include Policy Fragment** assertion into the service.
4. Set context variable for **apiLocation**.
   Repeat the same for Set context variable for **serviceUrl**.
5. Select **one** of the following fragments, depending on your method of API Protection:
   – Standard Policy Template Fragment - API Key
   – Standard Policy Template Fragment - OAuth 2.0
   – Standard Policy Template Fragment - No Auth



6. Delete **Route via http to ....**.

The analytics for the Gateway-published API is enabled.

View Analytics for a Gateway-Published API in API Explorer

Follow these steps:

1. In API Explorer, right-click the API and open **Published Service Properties**.

2. Under **HTTP/FTP > Allowed HTTP Methods**, enable **OPTIONS**.
3. Select **OK**.

Test your API and view the Analytics report. You should now be able to view API hits for your Gateway-published API.

Test and Explore APIs

As a Portal Admin or API owner, you can test and explore the APIs you published in API Management SaaS using the API Explorer or the Swagger UI. The API Explorer and Swagger UI only display APIs that have WADL or Swagger .json files.

> **NOTE**
> You can test and explore APIs using the API Explorer only by accessing it through the API Portal/Ingress tenant. If you are using an external tenant, test and explore APIs using the Swagger UI instead.

You can select an API and can configure a query using the API Explorer or Swagger UI. This function uses information in a well-crafted WADL or Swagger .json file and displays information about the API, resource, method, and parameters that you select for the query. A request is submitted against the API and a call is routed to the endpoint specified in the file. If multiple clusters of API proxies are present, you can point the endpoint that is specified in the WADL or Swagger file to a particular cluster or a virtual IP address.

> **NOTE**
> If the API that you are publishing is secured using OAuth 2.0, ensure that you specify the end point specified in the WADL or Swagger file in the **Spec Authentication** section of the API.
> For more information about how to configure the authentication settings for testing APIs, Create and Set Permissions for APIs.

When you send the query, the API Explorer or Swagger UI displays the query string and the API results. It can also display error messages.

You can translate the generated query into one of the following programming languages: cURL, Java, JavaScript, Node.js, Objective C, PHP, Python, and Ruby

Test and Explore API using API Explorer

Follow these steps:

1. Log in to API Portal as a Portal Admin or API Owner.
2. Select *one* of the following methods:
   - Use API Explorer. Select **Portal API**, **API Explorer**, and then select an API.
   - From the menu bar, select **Manage**, **APIs**, select the menu icon for the API that you want to test, and then select **Test**.
3. If the API requires authentication, perform the following substeps:
   a. From the **App/API Key** menu, select an application. The API Explorer uses the default API key and shared secret.
      > **NOTE**
      > When using the Portal API (PAPI) or Metrics API in the API Explorer, select **Portal API App**... from the **App/API Key** menu.
   b. Select **Configure Authentication**.
   c. Select the authentication type that the API requires.
      More fields appear. The fields depend on the authentication type.
   d. Complete the fields that the authentication type requires, and then select **OK**.
4. Select a resource in the API.
5. Select a method available for the resource. If the method requires a parameter value, then enter a valid value.

NOTE
API Portal supports **application/json** as the parameter content type. To view **application/xml** as the parameter type, manually convert the payload as an XML type, and paste as a body.

6. (Optional) If you want to add another parameter that the API supports, then perform the following substeps:
   a. Select **Add Parameter**.
   b. Specify the parameter type: query, header, or template.
      NOTE
      Template parameters are URI parameters that are enclosed in { }. Template parameters must be substituted before the URI can be resolved.
   c. Enter the parameter name and value, and then select **OK**.
7. Select **Submit**.
   The API Explorer displays the query that it sent to the API and the response from the API.
8. To translate the query to another programming language, select the language from the menu.

To go from the API Explorer to the API Portal Home page, select the name of your API Portal.

Test and Explore API using Swagger UI

Follow these steps:

1. While logged in to API Portal as a Portal Admin or API Owner, select **Manage**, **APIs**.
   A list of APIs display on the APIs page.
2. Select the API that you want to test and explore.
   The **Overview** tab opens.
3. Select the **Spec** tab.
   The Swagger UI opens.
4. From the drop-down, select the application for which you want to test and explore this API. If the API is not included in an application, you can click **Create an Application** to add one.
   A second drop-down appears for your to select your API key.
5. From the second drop-down, select your API key.
   The API key (client ID) and the shared secret (client secret) (if the shared secret was generated in plaintext), display.
6. Ensure that your session is authorized. If applicable, the *Padlock* icon next to your selected endpoint indicates whether an endpoint is locked. If required, authorize your session by clicking the **Padlock**, and then completing the information required in the *Authorization* window.
7. Expand the endpoint that you want to execute, and then click **Try it out**.
8. The example values in the *Request Body* field become editable. Make changes to the example request, and then click **Execute**.

Explore APIs

In this article:

Org Admins and Developers can read more about the APIs and can test the APIs that are published in API Portal using the API Explorer or the Swagger UI. API Explorer is accessible only through the API Portal/Ingress tenant. If you are using an external tenant, test and explore APIs using the Swagger UI instead. The API Explorer and Swagger UI display only APIs that have WADL or Swagger .json files.

You can select an API and configure a query using the API Explorer or Swagger UI. This function uses information in a well-crafted WADL or Swagger .json file and displays information about the API, resource, method, and parameters that you select for the query. A request is submitted against the API and a call is routed to the endpoint specified in the file. If multiple clusters of API proxies are present, you can point the endpoint that is specified in the WADL or Swagger file to a particular cluster or a virtual IP address.

You can translate the generated query into one of the following programming languages: Curl, Java, JavaScript, Node.js, Objective C, PHP, Python, and Ruby.

Test and Explore an API using API Explorer

Follow these steps:

1. Log in to API Portal as an Org Admin or Developer.
2. Select *one* of the following methods:
   - Use API Explorer. On the menu bar, select **API Explorer**, and then select an API.
   - On the menu bar, select **Manage**, **APIs**, select the menu icon for the API that you want to test, and then select **Test**.
3. If the API requires authentication, complete the following substeps:
   a. From the **App/API Key** menu, select an application. The API Explorer uses the default API key and shared secret.
      NOTE
      When using the Portal API (PAPI) or the Portal Metrics API in the API Explorer, select **Portal API app**.
   b. Select **Configure Authentication**.
   c. Select the authentication type that the API requires.
      More fields appear. The fields depend on the authentication type.
   d. Complete the fields that the authentication type requires, and then select **OK**.
4. Select a resource in the API.
5. Select a method available for the resource. If the method requires a parameter value, then enter a valid value.
   NOTE
   API Portal supports **application/json** as the parameter content type. To view **application/xml** as the parameter type, manually convert the payload as an XML type, and paste as a body.
6. (Optional) To add another parameter that the API supports, complete the following substeps:
   a. Select **Add Parameter**.
   b. Specify the parameter type: query, header, or template.
      NOTE
      Template parameters are URI parameters that are enclosed in { }. Template parameters must be substituted before the URI can be resolved.
   c. Enter the parameter name and value, and then select **OK**.
7. Select **Submit**.
   The API Explorer displays the query that it sent to the API and the response from the API.
8. To translate the query to another programming language, select the language from the menu.

To go from the API Explorer to the API Portal Home page, select the name of your API Portal.

Test and Explore an API using Swagger UI

Follow these steps:

1. While logged in to API Portal, from the menu bar, select **Manage**, **APIs**.
   The **APIs** page appears.
2. Select the tile for the API that you want to test and explore.
   The **Overview** tab opens.
3. Select the **Spec** tab.
   The Swagger UI opens.
4. From the drop-down, select the application for which you want to test and explore this API. If the API is not included in an application, you can click **Create an Application** to add one.
   A second drop-down appears for your to select your API key.
5. From the second drop-down, select your API key.

The API key (client ID) and the shared secret (client secret) (if the shared secret was generated in plaintext), display.

6. Ensure that your session is authorized. If applicable, the *Padlock* icon next to your selected endpoint indicates whether an endpoint is locked. If required, authorize your session by clicking the padlock, and then completing the information required in the *Authorization* window.
7. Expand your selected endpoint, and then click **Try it out**.
8. The example values in the *Request Body* field become editable. Make changes to the example request, and then click **Execute**.

Manage API Deployments

API Management SaaS provides targeted API deployments for deploying published APIs to specific proxies which represent a specific customer environment. You can manage multiple environments across organizations and functional or geographic locations using a single API Management SaaS to reduce the total cost of API ownership.

> **NOTE**
> You can also manage your API deployments by way of the Portal API (PAPI) or use this API in your scripts for managing API deployments.
>
> For more information on how to manage API deployments using PAPI, see Deployment API.

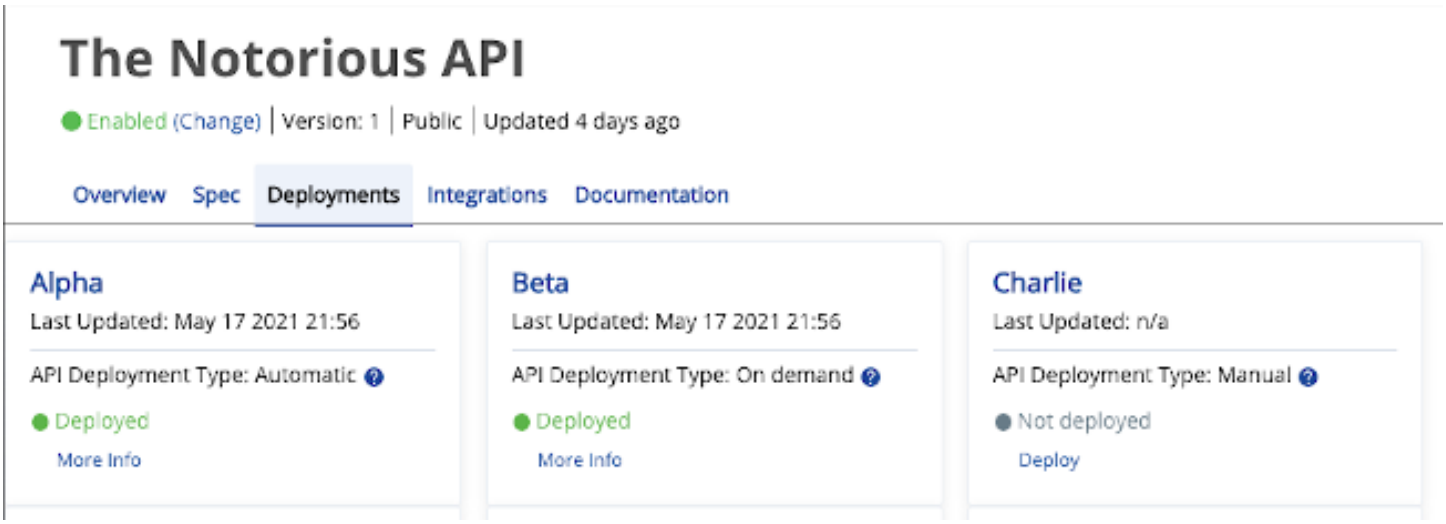
In this section:

API Deployment Types

Portal Admins and API Owners can control how APIs are deployed to a proxy using the API deployment type. You select the API deployment type when adding proxies.

For more information about how to add a proxy, see Integrate On-Premise API Proxies.

The API Details page displays the API deployment information on the **Deployments** tab. The following graphic displays the Details page of API deployment information for different API deployment types and proxies:



Proxies support the following API deployment types:

- **Automatic**

For this API deployment type, API Management SaaS immediately and automatically deploys changes to the published APIs to all proxies. For example, whenever an API is added, edited, or deleted. This is the default type.

- **On Demand**
  For this event-driven API deployment, users with deployment permissions can deploy the API to the proxy as needed. The Portal Deployer client handles deployment, undeployment, and redeployment of the API to the proxy. For example, this API deployment type is useful when you want to stagger the deployment of an API across all or your geographic regions to avoid impacting production instances during peak hours.
- **Scripted**
  For this API deployment type, you can integrate the API deployment into your CI/DC process by leveraging the Deployment API and invoking them from a custom API deployment script. The deployment APIs retrieve API deployment data and update the API deployment status for a proxy to keep API Management SaaS updated.

IMPORTANT

- To use the Automatic or On Demand API deployment types, you must install the latest Portal Integration bundle during the enrollment or upgrade process. You can check whether you have the latest bundle installed by viewing the Portal compatibility indicator as described in Manage Proxies.
- On Demand and scripted deployments are available only for Portal-published APIs.

The following sections describe each type in more detail.

Automatic API Deployment Type

Effect on API Deployments

Changes to APIs are automatically deployed to the proxy. For example, whenever an API is created, edited, or deleted.

To assist administrators in troubleshooting any synchronization of API changes between the API Portal and the proxy, the Proxy Details page lets you compare deployment data between proxy and Portal publishing sources and identify deployment errors immediately when they arise. New Gateway or proxy enrolments with the version 5.0.2 or newer with the latest Portal Integration bundle installed of the API Portal will take advantage of this synchronization feature. If your Portal installation uses an older Portal Integration bundle, this enhanced synchronization feature and Proxy Details page will not be available.

Use Case

The Automatic API deployment type is recommended for the following use cases:

- Rapid iteration of API development.
- Convenience and low maintenance.
- Development environments.
- You have Gateway published APIs.

Execution Method Used

UI and API Call

Notes

- The API count displays for the Portal and the proxy on the API Proxy Details page because the APIs are automatically deployed to the proxy.
- If you do not use the Automatic API deployment type for a proxy, the APIs are not synched to and from that proxy regardless of where they were created.
- Gateway-published APIs *must* use the Automatic API deployment type because they rely on the synchronized scheduled task to synchronize them from the proxy to the Portal.

CAUTION
If you have Gateway-published APIs and switch the proxy from Automatic to On Demand API deployment type, the counts on the Proxy Details page could be inaccurate since an On Demand API deployment type does not synchronize Gateway-published APIs.

To enable automatic deployment of Gateway-published APIs, you must enable migration between environments or clusters using Gateway Migration Utility (GMU). For more information, see the following links:

- Publish APIs with the API Portal
- Gateway Migration in the CA API Gateway documentation

On-Demand API Deployment Type

Effect on API Deployments

You can trigger API deployments on-demand by calling the Deployment API.

Deployments are triggered when a user with deployment permissions makes calls to the Apideployments resource in the Portal API (PAPI).

NOTE

On-demand deployments apply only to Portal-published APIs.

Use Case

The On Demand API deployment type is recommended for the following use cases:

- Deployments that are triggered as needed.
- Geographically distributed environments.
- UAT environments.

For example, you want to stagger the deployment of an API across all or your geographic regions to avoid impacting production instances during peak hours.

Execution Method Used

UI and API call.

Deployment APIs to orchestrate and trigger the deployment process.

Notes

The API Proxy Details page displays only the APIs that have been created on the Portal and not the APIs that have been deployed to a proxy.

Scripted API Deployment Type

Effect on API Deployments

You can integrate API deployments into your existing Continuous Integration/Continuous Deployment (CI/CD) process by using the deployment APIs and invoking them from your deployment script.

Deployments are triggered by the script in conjunction with the deployment APIs which are used to retrieve the API deployment data and update the API deployment status for a proxy.

NOTE
Scripted deployments apply only to Portal-published APIs.

Use Case

The scripted API deployment type is recommended for the following use cases:

- Integrations with existing CI/CD pipeline.
- Environments that are tightly controlled.
- Production environments.

For example, you want control over when an API is deployed using a custom script in conjunction with the Deployment APIs.

Execution Method Used

UI and API call.

Restman used to deploy to proxy.

Deployment APIs and custom script to orchestrate and trigger the deployment process.

Notes

API Proxy Details page displays only the APIs that have been created on the Portal and not the APIs that have been deployed to a proxy.

Deploy APIs

Publishers can deploy the APIs that they own to proxies or manage the proxies to which their organization has been assigned.

In this article:

Prerequisites

The following prerequisites apply when deploying APIs to proxies:

- **Management Permissions**: If you are an API Owner or Org Publisher deploying an API belonging to another Publisher, the Portal Admin or the default API Owner must give you permissions to manage that specific API.
- **Organizations Assignment**: If you are an Org Publisher, the Portal Admin must also assign your organization to the specific proxy.

Deploy an API to Proxies that Use the Automatic API Deployment Type

Portal-published APIs are automatically deployed to a proxy that uses the automatic API deployment type depending on the synchronization time. These API deployments are read-only and cannot be created, updated, or deleted.

Prerequisite: You have added the proxy using the steps listed in Integrate On-Premise API Proxies > Enroll Additional API Proxies, and you have selected the **Automatic** API deployment type.

Follow these steps:

1. From the menu bar, select **Manage**, **APIs**.
   A list of APIs display on the APIs page.
2. Create the API, which includes configuring the proxy URL. For more information, see Create and Set Permissions for APIs.

The API is deployed to the proxy.

Manage API Deployments to Proxies that Use the On Demand API Deployment Type

You can deploy, undeploy, or redeploy APIs to proxies that use the On Demand API deployment type using API Management SaaS, or by making calls to the `Deployments` resource for the Portal API (PAPI). The following procedure describes how to manage the deployment using API Management SaaS.

For more information about how to deploy APIs using PAPI, see Deploy APIs to Proxies using PAPI.

Follow these steps:

1. From the APIs page, open your API by selecting it.
   The **Overview** tab opens.
2. Select the **Deployments** tab.
3. In the selected proxy's card, select from the following:
   – If the proxy is not deployed, click **Deploy**.
   – If the proxy is already deployed or pending deployment, click **Redeploy** or **Undeploy**.
4. Click **Yes** when prompted.

Refresh the API Deployment Status to Proxies that Use the On Demand API Deployment Type

With your API open, on the **Deployments** tab, in your selected proxy's card, you can check the deployment status in the following ways:

- **Automatic:** After performing an action, the status refreshes in 10 seconds.
- **Manual:** Select **Check now** anytime while the action is in progress.

Deploy an API to Proxies that Use the Scripted API Deployment Type

You manage API deployment to proxies that use the scripted API deployment type (deploy, undeploy, or redeploy) by making calls to the `Deployments` resource for the PAPI.

Deploy APIs to Proxies using PAPI

You can deploy APIs to proxies by making calls to the `Deployments` resource for the Portal API (PAPI) for the following purposes:

- To deploy APIs to proxies with the On Demand or Scripted API deployment type (On Demand or Scripted proxies). For proxies with the Automatic API deployment type, the deployments are read-only and cannot be created, updated, or deleted.
- To assign organizations to a proxy.

  NOTE

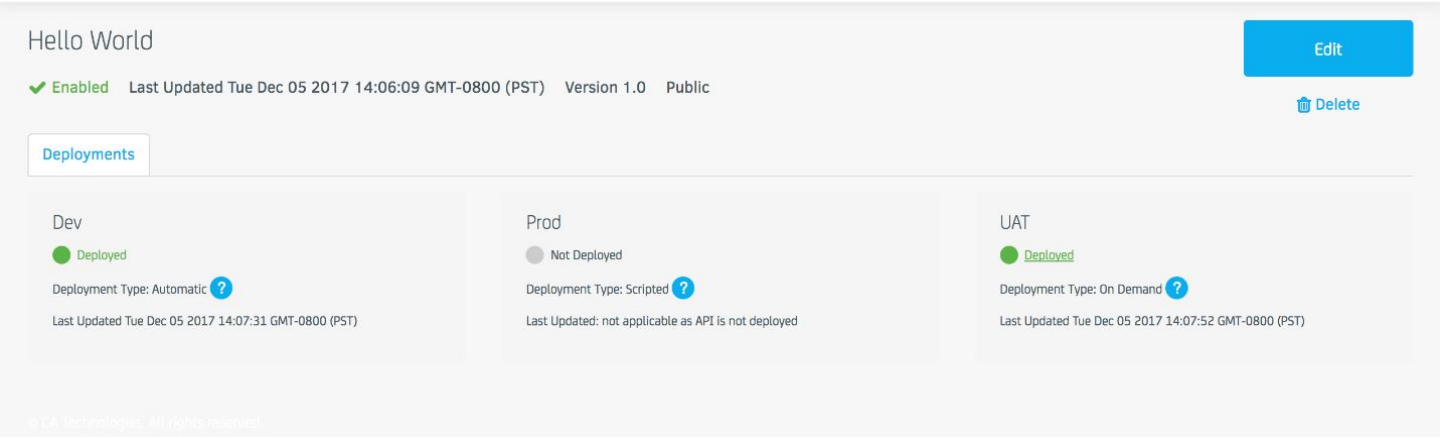
  - (Automatic and On Demand proxies only) To deploy APIs to proxies using API Portal, see Deploy APIs.
  - To assign organizations to a proxy using API Portal, see Manage Proxies.

Prerequisites

- You have retrieved a valid OAuth 2.0 access token. The API calls mentioned are also documented through the API Explorer accessible through the Portal API's functionality.
- You have installed the REST Management API (restman) on your proxy:
  – For more information, see Install the REST Management Service in the API Gateway documentation.
  – For the SaaS solution, restman is installed for you.

  NOTE
  API Explorer is only accessible through the API Portal/Ingress tenant.

You can manage API deployments to proxies by making calls to the `Deployments` resource for the PAPI. The following example displays how to use the deployments resource for three active proxies:



- DEV proxy with the AUTOMATIC API deployment type, which maps to API Portal as automatic.
- UAT proxy with the ON_DEMAND API deployment type, which maps to API Portal as on demand.
- PROD proxy with a MANUAL API deployment type, which maps to API Portal as scripted.

Retrieve the list of proxies using the following command:

```
curl -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/deployments/1.0/proxies
```

Example request:

```
curl -H 'Authorization: Bearer cde69bcc-3bed-44e0-af5b-c33fcb9020d5' \
https://apim-ssg-apim-uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/proxies
```

Example response:

```
[
    {
        "uuid": "8f6bc46c-a131-4388-b654-1e2b599b0ee9",
        "name": "DEV",
        "enrollmentStatus": "ACTIVE",
        "deploymentType": "AUTOMATIC"
    },
    {
        "uuid": "de484ed2-cea4-4adc-885b-bf495f94b9f7",
        "name": "UAT",
        "enrollmentStatus": "ACTIVE",
        "deploymentType": "ON_DEMAND"
    },
    {
        "uuid": "639536fc-230d-434e-8b88-3e13d5069c34",
        "name": "PROD",
        "enrollmentStatus": "ACTIVE",
        "deploymentType": "MANUAL"
    }
]
```

From the response example, you can see that there are three proxies currently enrolled with the Portal: DEV, UAT, PROD. Each proxy has a different API deployment type and one of them has API deployments that are handled automatically. For the other two proxies (On Demand and Scripted), you manage API deployments to proxies by making calls to the `Deployments` resource for the PAPI.

NOTE

- Assign Proxy Organizations using Deployment API
- Deploy Proxy with On-Demand Deployments
- Deploy Proxy with Scripted Deployments

Manage Organization Assignments to Proxies using PAPI

As a Portal Admin, you can manage organization assignments to proxies using the `Proxies` resource for the Portal API (PAPI). Requests for organization assignment are handled by way of calls to the `/deployments/1.0/proxies/{uuid}/organizations` endpoint.

Retrieve the Organizations Assigned to a Proxy

Issue the following command:

```
curl -H "Authorization: Bearer {token}" https://{portalApiHost}/{tenantId}/deployments/1.0/proxies/
{proxyUuid}/organizations
```

Example request:

```
curl -H "Authorization:Bearer 5d9646b0-6290-412c-98c3-82cba4c88fce" -request GET "https://apim-ssg-apim-
uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/proxies/d55b86a3-e5ab-48e6-8762-7078ea23bc83/
organizations

for local environment:
curl --include   --header "Authorization:Bearer 70f8c255-8d0c-4362-95e9-652569c6e9e0" \
--request GET "https://apim-ssg.dev.ca.com:9443/t1/deployments/1.0/d55b86a3-e5ab-48e6-8762-7078ea23bc83/
organizations"
```

Example response:

```
[
  {
    "uuid": "2c7f1bbd-c79d-4202-83c8-e8aa7f8ce565"
    "name": "Organization A"
  },
  {
    "uuid": "de484ed2-cea4-4adc-885b-bf495f94b9f7"
    "name": "Organization B"
  }
]
```

Assign an Organization to a Proxy

Issue the following command:

```
curl -X PUT -H 'Authorization: Bearer {token}' -H 'Content-Type:application/json;charset=utf-8' \ https://
{portalApiHost}/{tenantId}/deployments/1.0/proxies/{proxyUuid}/organizations \ -d '{"orgUuid": "{orgUuid}"}'
```

Example request:

```
curl -X POST -H 'Authorization: Bearer 5d9646b0-6290-412c-98c3-82cba4c88fce' -H 'Content-Type:application/
json;charset=utf-8' \ https://apim-ssg-apim-uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/
proxies/d55b86a3-e5ab-48e6-8762-7078ea23bc83/organizations \ -d '{"orgUuid": "2c7f1bbd-c79d-4202-83c8-
e8aa7f8ce565"}'

for local environment:
curl --include --header "Authorization:Bearer 70f8c255-8d0c-4362-95e9-652569c6e9e0" \
```

```
  --data '2c7f1bbd-c79d-4202-83c8-e8aa7f8ce565' \
  --request PUT "https://apim-ssg.dev.ca.com:9443/t1/deployments/1.0/proxies/d55b86a3-
e5ab-48e6-8762-7078ea23bc83/organizations"
```

Unassign an Organization from a Proxy

You unassign organizations from proxies by editing the proxy using API Portal. For more information about how to edit proxies, see Manage Proxies.

Deploy Proxy with On-Demand API Deployments

For proxies with the on-demand API deployment type, users with deployment permissions can deploy APIs to the proxy as needed. The Portal Deployer client handles the deployment, undeployment, and redeployment of APIs to the proxy.

NOTE

On-demand API deployments are only available for Portal-published APIs.

The following information are examples of on-demand API deployment calls:

- REST API On-Demand API Deployment
- SOAP API On-Demand API Deployment

NOTE

- Manage API Deployments
- Deployment API

REST API On-Demand Deployment

This topic includes the workflow for deploying a REST API on demand.

View the Current Deployment of an API

Use the following command:

```
curl -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/deployments/1.0/apis/{apiUuid}/
proxies
```

Example request:

```
curl -H 'Authorization: Bearer cde69bcc-3bed-44e0-af5b-c33fcb9020d5' \
https://apim-ssg-apim-uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/
apis/430924a3-d279-4d42-9eec-46da03ea3846/proxies
```

Example response:

```
[
    {
        "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
        "proxyUuid": "8f6bc46c-a131-4388-b654-1e2b599b0ee9",
        "proxyName": "DEV",
        "lastTimeDeployed": 1511994241499,
        "status": "DEPLOYED"
    }
]
```

In this example, the API with uuid 43e52d60-d4e3-4c15-89c1-8764de4e3106 has been deployed to the DEV proxy automatically.

Create an API Deployment

Creating an API deployment deploys an API to the UAT proxy that uses the On Demand API deployment type. Issue the following command:

```
curl -X POST -H 'Authorization: Bearer {token}' -H 'Content-Type:application/json;charset=utf-8' \
https://{portalApiHost}/{tenantId}/deployments/1.0/apis/{apiUuid}/proxies \
-d '{"proxyUuid": "{proxyUuid}"}'
```

Example request:

```
curl -X POST -H 'Authorization: Bearer cde69bcc-3bed-44e0-af5b-c33fcb9020d5' -H 'Content-Type:application/
json;charset=utf-8' \
https://apim-ssg-apim-uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/apis/430924a3-
d279-4d42-9eec-46da03ea3846/proxies \
-d '{"proxyUuid": "de484ed2-cea4-4adc-885b-bf495f94b9f7"}'
```

Example response:

```
{
    "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
    "proxyUuid": "de484ed2-cea4-4adc-885b-bf495f94b9f7",
    "proxyName": "UAT",
    "lastTimeDeployed": 0,
    "status": "PENDING_DEPLOYMENT"
}
```

When you create the API deployment, API Management SaaS publishes a deployment event for the UAT proxy Portal Deployer client to consume. If you make the previous API call to retrieve an API deployment, the following information appears:


```
[
    {
        "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
        "proxyUuid": "8f6bc46c-a131-4388-b654-1e2b599b0ee9",
        "proxyName": "DEV",
        "lastTimeDeployed": 1511994960953,
        "status": "DEPLOYED"
    },
    {
        "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
        "proxyUuid": "de484ed2-cea4-4adc-885b-bf495f94b9f7",
        "proxyName": "UAT",
        "lastTimeDeployed": 1511994641295,
        "status": "DEPLOYED",
        "message": "[{\"targetLocation\":\"https://localhost:8443/restman/1.0/bundle\",\"message\":
\"<?xml version=\\\"1.0\\\" encoding=\\\"UTF-8\\\" standalone=\\\"yes\\\"?>\\n<l7:Item xmlns:l7=\\
\"http://ns.l7tech.com/2010/04/gateway-management\\\">\\n    <l7:Name>Bundle mappings</l7:Name>\\n
 <l7:Type>BUNDLE MAPPINGS</l7:Type>\\n    <l7:TimeStamp>2017-11-29T22:30:41.069Z</l7:TimeStamp>\\n
 <l7:Link rel=\\\"self\\\" uri=\\\"https://localhost:8443/restman/1.0/bundle\\\"/>\\n    <l7:Resource>\
\n        <l7:Mappings>\\n            <l7:Mapping action=\\\"NewOrUpdate\\\" actionTaken=\\\"CreatedNew
\\\" srcId=\\\"43e52d60d4e34c1589c18764de4e3106\\\" targetId=\\\"43e52d60d4e34c1589c18764de4e3106\\\"
 targetUri=\\\"https://localhost:8443/restman/1.0/policies/43e52d60d4e34c1589c18764de4e3106\\\" type=\\
\"POLICY\\\"/>\\n            <l7:Mapping action=\\\"NewOrUpdate\\\" actionTaken=\\\"CreatedNew\\\" srcId=
```

```
\\\"144251a872943a3a8be915f4eaf1e69f\\\" targetId=\\\"144251a872943a3a8be915f4eaf1e69f\\\" targetUri=\\
\"https://localhost:8443/restman/1.0/services/144251a872943a3a8be915f4eaf1e69f\\\" type=\\\"SERVICE\\\"/>\\n
      </l7:Mappings>\\n    </l7:Resource>\\n</l7:Item>\\n\",\"status\":\"DEPLOYED\"}]"
    }
]
```

From the response example, the Portal Deployer client has deployed the API to the UAT proxy and has set its status to DEPLOYED from PENDING_DEPLOYMENT. It has set lastTimeDeployed. If an error occurred during the API deployment, the Portal Deployer client sets the status to ERROR instead and the message field contains the error. If the Portal Deployer client has not deployed the API to the proxy and the API deployment itself is stuck in PENDING_DEPLOYMENT, there might be a connectivity issue between the Portal Deployer client and API Management SaaS. For more information about issues with the Portal Deployer client, see Troubleshoot API Deployments.

Redeploy the API

The following API properties can impact API deployment to a proxy when changed:

- name
- state
- policy templates
- policy templates values
- custom fields
- custom fields values

The Portal Deployer client sets back the API deployment status to PENDING_DEPLOYMENT to denote to the user that the API *must* be redeployed, as the version deployed on the proxy is no longer the same as the one published in API Management SaaS. Update the API deployment to `PENDING_DEPLOYMENT` by issuing the following command:

```
curl -X PUT -H 'Authorization: Bearer {token}' -H 'Content-Type:application/json;charset=utf-8' \
https://{portalApiHost}/{tenantId}/deployments/1.0/apis/{apiUuid}/proxies/{proxyUuid} \
-d '{"status": "PENDING_DEPLOYMENT"}'
```

Example request:

```
curl -X PUT -H 'Authorization: Bearer cde69bcc-3bed-44e0-af5b-c33fcb9020d5' -H 'Content-Type:application/
json;charset=utf-8' \
https://apim-ssg-apim-uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/apis/430924a3-
d279-4d42-9eec-46da03ea3846/proxies/de484ed2-cea4-4adc-885b-bf495f94b9f7 \
-d '{"status": "PENDING_DEPLOYMENT"}'
```

Example response:

```
{
    "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
    "proxyUuid": "de484ed2-cea4-4adc-885b-bf495f94b9f7",
    "proxyName": "UAT",
    "lastTimeDeployed": 1512000698619,
    "status": "PENDING_DEPLOYMENT"
}
```

Retrieving the API deployments, you can see that the following API was deployed again and has a new lastTimeDeployed:

```
[
    {
        "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
        "proxyUuid": "8f6bc46c-a131-4388-b654-1e2b599b0ee9",
        "proxyName": "DEV",
        "lastTimeDeployed": 1512000901396,
```

```
        "status": "DEPLOYED"
    },
    {
        "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
        "proxyUuid": "de484ed2-cea4-4adc-885b-bf495f94b9f7",
        "proxyName": "UAT",
        "lastTimeDeployed": 1512000847136,
        "status": "DEPLOYED",
        "message": "[{\"targetLocation\":\"https://localhost:8443/restman/1.0/bundle\",\"message\":\"<?
xml version=\\\"1.0\\\" encoding=\\\"UTF-8\\\" standalone=\\\"yes\\\"?>\\n<l7:Item xmlns:l7=\\\"http://
ns.l7tech.com/2010/04/gateway-management\\\">\\n    <l7:Name>Bundle mappings</l7:Name>\\n    <l7:Type>BUNDLE
 MAPPINGS</l7:Type>\\n    <l7:TimeStamp>2017-11-30T00:14:06.916Z</l7:TimeStamp>\\n    <l7:Link rel=\\\"self
\\\" uri=\\\"https://localhost:8443/restman/1.0/bundle\\\"/>\\n    <l7:Resource>\\n        <l7:Mappings>
\\n            <l7:Mapping action=\\\"NewOrUpdate\\\" actionTaken=\\\"UpdatedExisting\\\" srcId=\\
\"43e52d60d4e34c1589c18764de4e3106\\\" targetId=\\\"43e52d60d4e34c1589c18764de4e3106\\\" targetUri=\
\\"https://localhost:8443/restman/1.0/policies/43e52d60d4e34c1589c18764de4e3106\\\" type=\\\"POLICY\\
\"/>\\n            <l7:Mapping action=\\\"NewOrUpdate\\\" actionTaken=\\\"UpdatedExisting\\\" srcId=\
\\"144251a872943a3a8be915f4eaf1e69f\\\" targetId=\\\"144251a872943a3a8be915f4eaf1e69f\\\" targetUri=\\
\"https://localhost:8443/restman/1.0/services/144251a872943a3a8be915f4eaf1e69f\\\" type=\\\"SERVICE\\\"/>\\n
      </l7:Mappings>\\n    </l7:Resource>\\n</l7:Item>\\n\",\"status\":\"DEPLOYED\"}]"
    }
]
```


Undeploy the API

You undeploy an API by deleting it from the proxy. Issue the following command:

```
curl -X DELETE -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/deployments/1.0/apis/
{apiUuid}/proxies/{proxyUuid}
```

Example request:

```
curl -X DELETE -H 'Authorization: Bearer cde69bcc-3bed-44e0-af5b-c33fcb9020d5' \
https://apim-ssg-apim-uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/apis/430924a3-
d279-4d42-9eec-46da03ea3846/proxies/de484ed2-cea4-4adc-885b-bf495f94b9f7
```

When deleting an API from the proxy, the following events occur:

- The API state is set to PENDING_UNDEPLOYMENT.
- API Management SaaS publishes an undeploy event.

If the the Portal Deployer client successfully processes the undeploy event on the proxy, it deletes the API from the proxy and deletes the API deployment from API Management SaaS. If deleting the API fails, the Portal Deployer client updates the API deployment with an ERROR state and the message field contains the actual error. If the API deployment is stuck in PENDING_UNDEPLOYMENT, there might be a connectivity issue between the Portal Deployer client and API Management SaaS. For more information, see Troubleshoot API Deployments. If successful, retrieving the API deployments displays that the API deployment as deleted:

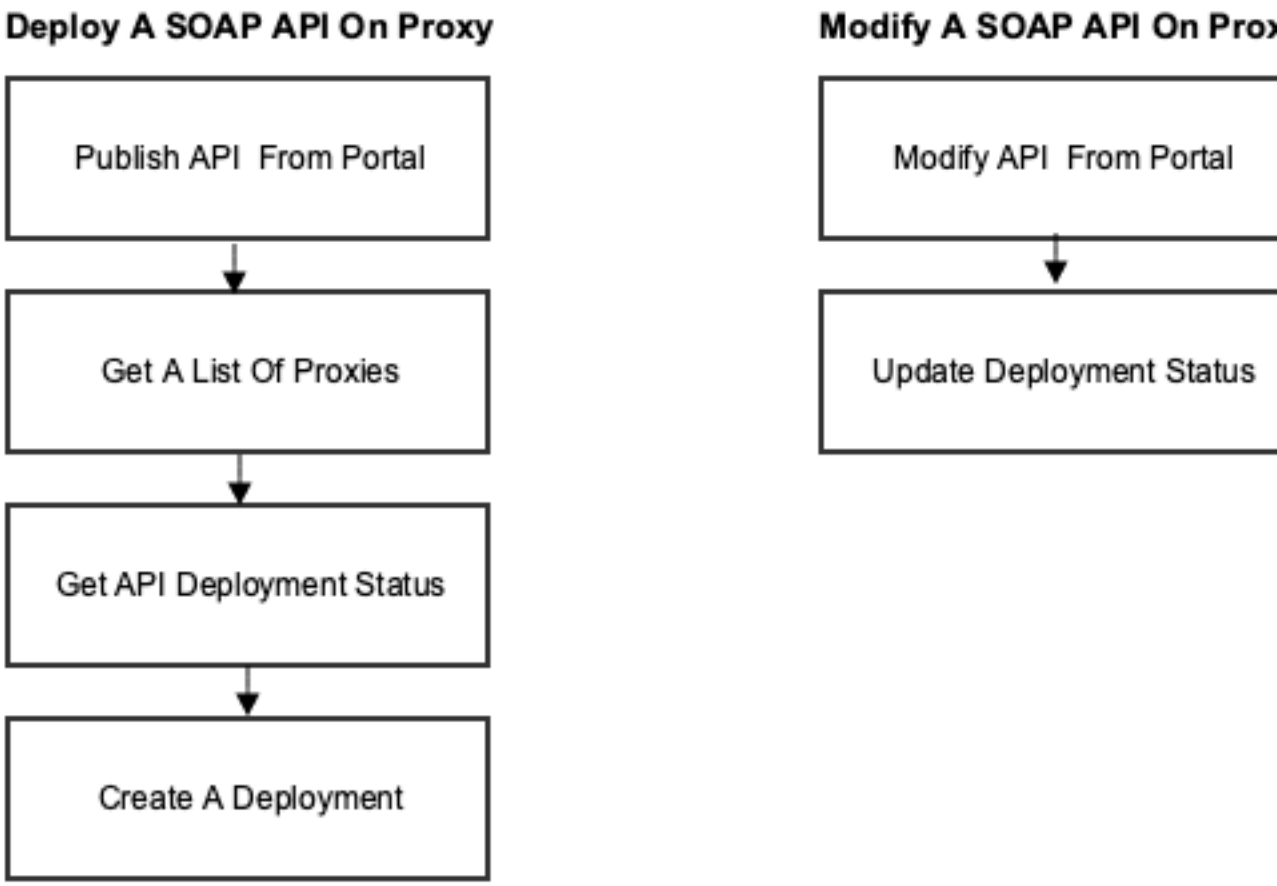

```
[
    {
        "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
        "proxyUuid": "8f6bc46c-a131-4388-b654-1e2b599b0ee9",
        "proxyName": "DEV",
        "lastTimeDeployed": 1512001260997,
        "status": "DEPLOYED"
    }
```

```
]
```

SOAP API On-Demand API Deployment

This topic includes information about how to deploy, modify, and undeploy a SOAP API on demand.

The workflows are as follows:



Deploy a SOAP API On Demand

Follow these steps:

1. Get a list of proxies by issuing the following command:

```
curl -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/
deployments/1.0/proxies
```

Example request

```
curl -H "Authorization:Bearer 5d9646b0-6290-412c-98c3-82cba4c88fce" --request GET "https://apim-ssg-
soapsupport.app.dev1.w2.saasqa.ca.com:443/tenantsoapsupport/deployments/1.0/proxies

for local environment:
curl --include   --header "Authorization:Bearer 70f8c255-8d0c-4362-95e9-652569c6e9e0" \
  --request GET "https://apim-ssg.dev.ca.com:9443/t1/deployments/1.0/proxies"
```

Example response

```
[
   {
      "uuid":"845b69bc-0aa6-425c-91d3-99a37bca0221",
      "name":"UAT",
      "enrollmentStatus":"ACTIVE",
      "deploymentType":"ON_DEMAND"
   }
]
```

```
[
    {
        "uuid": "f5fa57da-2604-43d3-8a58-02d1c61be945",
        "name": "t1_proxy_automatic",
        "enrollmentStatus": "ACTIVE",
        "deploymentType": "AUTOMATIC"
    },
    {
        "uuid": "a94a1f06-04c4-4229-842b-a087722bb4c9",
        "name": "t1_proxy_ondemand",
        "enrollmentStatus": "ACTIVE",
        "deploymentType": "ON_DEMAND"
    },
    {
        "uuid": "47b6f8f1-83dc-42df-bcfc-c8208bc5782e",
        "name": "t1_proxy_scripted",
        "enrollmentStatus": "ACTIVE",
        "deploymentType": "MANUAL"
    }
]
```


2. Get API deployment status by issuing the following command:

```
curl -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/deployments/1.0/apis/{apiUuid}/
proxies
```

Example request

```
https://apim-ssg-soapsupport.app.dev1.w2.saasqa.ca.com:443/tenantsoapsupport/deployments/1.0/apis/
d77f1e16-c899-4556-8a16-09c283cb8f8d/proxies

for local environment:
curl --include   --header "Authorization:Bearer 70f8c255-8d0c-4362-95e9-652569c6e9e0" \
  --request GET "https://apim-ssg.dev.ca.com:9443/t1/deployments/1.0/apis/5cf8ba84-cef9-41e5-9b75-
e06979337649/proxies"
```

Example response


```
[]
```

```
[
    {
        "apiUuid": "5cf8ba84-cef9-41e5-9b75-e06979337649",
        "proxyUuid": "f5fa57da-2604-43d3-8a58-02d1c61be945",
        "proxyName": "t1_proxy_automatic",
        "lastTimeDeployed": 1562308560417,
        "status": "DEPLOYED"
```

```
    }
]
```

3. Create a deployment by issuing the following command:

```
curl -X POST -H 'Authorization: Bearer {token}' -H 'Content-Type:application/json;charset=utf-8' \
https://{portalApiHost}/{tenantId}/deployments/1.0/apis/{apiUuid}/proxies \
-d '{"proxyUuid": "{proxyUuid}"}'
```

Example request

```
curl -X POST -H 'Authorization: Bearer 5d9646b0-6290-412c-98c3-82cba4c88fce' -H 'Content-Type:application/
json;charset=utf-8' \
https://apim-ssg-soapsupport.app.dev1.w2.saasqa.ca.com:443/tenantsoapsupport/deployments/1.0/apis/
d77f1e16-c899-4556-8a16-09c283cb8f8d/proxies \
-d '{"proxyUuid": "845b69bc-0aa6-425c-91d3-99a37bca0221"}'

for local environment:
 curl --include   --header "Authorization:Bearer 70f8c255-8d0c-4362-95e9-652569c6e9e0"  --header "Content-
Type:application/json;charset=UTF-8" \
  --data '{
  "proxyUuid": "a94a1f06-04c4-4229-842b-a087722bb4c9"
}' \
  --request POST "https://apim-ssg.dev.ca.com:9443/t1/deployments/1.0/apis/5cf8ba84-cef9-41e5-9b75-
e06979337649/proxies"
```

Example response

```
{
   "apiUuid":"d77f1e16-c899-4556-8a16-09c283cb8f8d",
   "proxyUuid":"845b69bc-0aa6-425c-91d3-99a37bca0221",
   "proxyName":"UAT",
   "lastTimeDeployed":0,
   "status":"PENDING_DEPLOYMENT"
}



{
    "apiUuid": "5cf8ba84-cef9-41e5-9b75-e06979337649",
    "proxyUuid": "a94a1f06-04c4-4229-842b-a087722bb4c9",
    "proxyName": "t1_proxy_ondemand",
    "lastTimeDeployed": 0,
    "status": "PENDING_DEPLOYMENT"
          }
```

Eventually, the status is updated to "DEPLOYED".

Modify an On Demand API Deployment

Issue the following command:

```
curl -X PUT -H 'Authorization: Bearer {token}' -H 'Content-Type:application/json;charset=utf-8' \
https://{portalApiHost}/{tenantId}/deployments/1.0/apis/{apiUuid}/proxies/{proxyUuid} \
-d '{"status": "PENDING_DEPLOYMENT"}'
```

Example request:

```
curl -X PUT -H 'Authorization: Bearer aa221b8d-5101-428c-b6cc-c9a2fe94f2b0' -H 'Content-Type:application/
json;charset=utf-8' \
```

```
https://apim-ssg-soapsupport.app.dev1.w2.saasqa.ca.com:443/tenantsoapsupport/deployments/1.0/apis/d77f1e16-
c899-4556-8a16-09c283cb8f8d/proxies/845b69bc-0aa6-425c-91d3-99a37bca0221 \
-d '{"status": "PENDING_DEPLOYMENT"}'


for local environment:
 curl --include   --header "Authorization:Bearer 70f8c255-8d0c-4362-95e9-652569c6e9e0"  --header "Content-
Type:application/json;charset=UTF-8" \
  --data '{
  "message": "string",
  "status": "PENDING_DEPLOYMENT"
}' \
  --request PUT "https://apim-ssg.dev.ca.com:9443/t1/deployments/1.0/apis/5cf8ba84-cef9-41e5-9b75-
e06979337649/proxies/a94a1f06-04c4-4229-842b-a087722bb4c9"
```

Example response:

```
{
   "apiUuid":"d77f1e16-c899-4556-8a16-09c283cb8f8d",
   "proxyUuid":"845b69bc-0aa6-425c-91d3-99a37bca0221",
   "proxyName":"UAT",
   "lastTimeDeployed":1557252051085,
   "status":"PENDING_DEPLOYMENT"
}

{
    "apiUuid": "5cf8ba84-cef9-41e5-9b75-e06979337649",
    "proxyUuid": "a94a1f06-04c4-4229-842b-a087722bb4c9",
    "proxyName": "t1_proxy_ondemand",
    "lastTimeDeployed": 1562310149900,
    "status": "PENDING_DEPLOYMENT",
    "message": "string"
}
```

Eventually, the status is updated to "DEPLOYED".

Undeploy an On Demand Deployment

Follow these steps:

1. Retrieve the API delete bundle by issuing the following command:

   ```
   curl -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/api-
   management/1.0/apis/{apiUuid}/bundle?type=delete
   ```

   Example request:

   ```
   curl -k -H 'Authorization: Bearer 1313cea7-fc46-41d0-b983-b4a717121cb2' https://apim-ssg-
   soapsupport.app.dev1.w2.saasqa.ca.com:443/tenantsoapsupport/api-management/1.0/apis/d77f1e16-
   c899-4556-8a16-09c283cb8f8d/bundle?type=delete \
   -o soap-api-bundle_delete.xml
   ```

   Content of soap-api-bundle_delete.xml

   ```
   <l7:Bundle xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
       <l7:Mappings>
               <l7:Mapping action="Delete" srcId="b2286ad6a8243794af20976095151300" type="SERVICE"
    xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management"></l7:Mapping>
   ```

```
            <l7:Mapping action="Delete" srcId="d77f1e16c89945568a1609c283cb8f8d" type="POLICY"
 xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management"></l7:Mapping>
        </l7:Mappings>
</l7:Bundle>
```

2. Delete the API on the proxy by issuing the following command:

```
curl -X PUT -H 'Content-Type: application/xml' \
-u {username}:{password} -d @api-bundle.xml \
https://{proxyHost}/restman/1.0/bundle
```

Example request:

```
curl -X PUT -H 'Content-Type: application/xml' \
-u admin:password -d @soap-api-bundle_delete.xml \
https://ohgateway1-94.lvn.broadcom.net:8443/restman/1.0/bundle
```

```
 curl --include   --header "Authorization:Bearer 70f8c255-8d0c-4362-95e9-652569c6e9e0" \
  --request DELETE "https://apim-ssg.dev.ca.com:9443/t1/Apis('5cf8ba84-cef9-41e5-9b75-e06979337649')"
```

Example response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<l7:Item xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
    <l7:Name>Bundle mappings</l7:Name>
    <l7:Type>BUNDLE MAPPINGS</l7:Type>
    <l7:TimeStamp>2019-05-06T11:53:47.083-07:00</l7:TimeStamp>
    <l7:Link rel="self" uri="https://ohgateway1-94.lvn.broadcom.net:8443/restman/1.0/bundle"/>
    <l7:Resource>
        <l7:Mappings>
            <l7:Mapping action="Delete" actionTaken="Deleted" srcId="b2286ad6a8243794af20976095151300"
 targetId="b2286ad6a8243794af20976095151300" targetUri="https://ohgateway1-94.lvn.broadcom.net:8443/
restman/1.0/services/b2286ad6a8243794af20976095151300" type="SERVICE"/>
            <l7:Mapping action="Delete" actionTaken="Deleted" srcId="d77f1e16c89945568a1609c283cb8f8d"
 targetId="d77f1e16c89945568a1609c283cb8f8d" targetUri="https://ohgateway1-94.lvn.broadcom.net:8443/
restman/1.0/policies/d77f1e16c89945568a1609c283cb8f8d" type="POLICY"/>
        </l7:Mappings>
    </l7:Resource>
</l7:Item>
```

3. Delete the API deployment by issuing the following command:

```
curl -X DELETE -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/deployments/1.0/apis/
{apiUuid}/proxies/{proxyUuid}
```

Example request:

```
curl -X DELETE -H 'Authorization: Bearer 1313cea7-fc46-41d0-b983-b4a717121cb2' \
https://apim-ssg-soapsupport.app.dev1.w2.saasqa.ca.com:443/tenantsoapsupport/deployments/1.0/apis/
d77f1e16-c899-4556-8a16-09c283cb8f8d/proxies/845b69bc-0aa6-425c-91d3-99a37bca0221
```

Deploy Proxy with Scripted Deployments

For proxies with the scripted API deployment type, the deployment, undeployment, and redeployment of APIs to the proxy is performed manually by a process outside of the control of API Portal. This deployment type is ideal if you want explicit control of proxies and you do not want API Portal to automatically deploy your APIs to the proxy.

> **NOTE**
> Scripted deployment is only available for API Portal-published APIs

The following articles are examples of scripted API deployment calls:

- REST API Scripted Deployment
- SOAP API Scripted Deployment

NOTE

- Manage API Deployments
- Deployment API

REST API Scripted Deployment

The following is an example workflow of REST API scripted deployment calls.

View the API's Current Deployments

Use the following command:

```
curl -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/
deployments/1.0/apis/{apiUuid}/proxies
```

Example request

```
curl -H 'Authorization: Bearer cde69bcc-3bed-44e0-af5b-c33fcb9020d5' \ https://apim-ssg-apim-uswest-
prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/apis/43e52d60-d4e3-4c15-89c1-8764de4e3106/proxies
```

Example response

```
[
    {
        "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
        "proxyUuid": "8f6bc46c-a131-4388-b654-1e2b599b0ee9",
        "proxyName": "DEV",
        "lastTimeDeployed": 1511994241499,
        "status": "DEPLOYED"
    }
]
```

From the response, you can see that the API with uuid `43e52d60-d4e3-4c15-89c1-8764de4e3106` was deployed to the DEV proxy automatically.

Create a Deployment

To deploy this API to the PROD proxy, create a new API deployment using the following command:

```
curl -X POST -H 'Authorization: Bearer {token}' -H 'Content-Type:application/json;charset=utf-8' \ https://
{portalApiHost}/{tenantId}/deployments/1.0/apis/{apiUuid}/proxies \ -d '{"proxyUuid": "{proxyUuid}"}'
```

Example request

```
curl -X POST -H 'Authorization: Bearer 860c9661-3e83-4672-ad36-ac8ea65c639b' -H 'Content-Type:application/
json;charset=utf-8' \ https://apim-ssg-apim-uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/
apis/43e52d60-d4e3-4c15-89c1-8764de4e3106/proxies \ -d '{"proxyUuid": "639536fc-230d-434e-8b88-3e13d5069c34"}'
```

Example response

```
{
    "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
    "proxyUuid": "639536fc-230d-434e-8b88-3e13d5069c34",
    "proxyName": "PROD",
    "lastTimeDeployed": 0,
    "status": "PENDING_DEPLOYMENT"
}
```

Because this proxy is configured with scripted deployment type, the API is manually deployed to the proxy and the deployment has a state of PENDING_DEPLOYMENT until it is manually updated. If you make the previous API call to retrieve an API deployments, you see the following information:

```
[
    {
        "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
        "proxyUuid": "8f6bc46c-a131-4388-b654-1e2b599b0ee9",
        "proxyName": "DEV",
        "lastTimeDeployed": 1512077761305,
        "status": "DEPLOYED"
    },
    {
        "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
        "proxyUuid": "639536fc-230d-434e-8b88-3e13d5069c34",
        "proxyName": "PROD",
        "lastTimeDeployed": 0,
        "status": "PENDING_DEPLOYMENT"
    }
]
```


Retrieve the API Bundle

To deploy the API to the proxy manually, retrieve the API bundle from the Portal by making the following request:

```
curl -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/api-management/1.0/apis/{apiUuid}/
bundle
```

Example request

```
curl -H 'Authorization: Bearer 14f1f709-1b19-4c29-a0fc-c93cdc683159' \ https://apim-ssg-apim-uswest-
prod.app.prod.w2.dev.ca.com:443/atenant/api-management/1.0/apis/43e52d60-d4e3-4c15-89c1-8764de4e3106/bundle \
 -o api-bundle.xml
```

Example response, saved to api-bundle.xml:


```
<l7:Bundle
    xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
    <l7:References>
        <l7:Item
            xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
            <l7:Name>Demo Api 294ece4a - updated-fragment</l7:Name>
            <l7:Id>43e52d60d4e34c1589c18764de4e3106</l7:Id>
            <l7:Type>POLICY</l7:Type>
            <l7:Resource>
                <l7:Policy guid="43e52d60-d4e3-4c15-89c1-8764de4e3106" id="43e52d60d4e34c1589c18764de4e3106"
 version="0">
                    <l7:PolicyDetail folderId="ddb84c6f397d7dbd3cca71d3043f019c" guid="43e52d60-
d4e3-4c15-89c1-8764de4e3106" id="43e52d60d4e34c1589c18764de4e3106">
                        <l7:Name>Demo Api 294ece4a - updated-fragment</l7:Name>
                        <l7:PolicyType>Include</l7:PolicyType>
                        <l7:Properties>
                            <l7:Property key="revision">
                                <l7:LongValue>1</l7:LongValue>
                            </l7:Property>
                            <l7:Property key="soap">
                                <l7:BooleanValue>false</l7:BooleanValue>
                            </l7:Property>
                        </l7:Properties>
                    </l7:PolicyDetail>
                    <l7:Resources>
                        <l7:ResourceSet tag="policy">
```

```
                            <l7:Resource type="policy">&lt;?xml version=&quot;1.0&quot;
 encoding=&quot;UTF-8&quot;?&gt;
&lt;wsp:Policy
                                xmlns:L7p=&quot;http://www.layer7tech.com/ws/policy&quot;
                                xmlns:wsp=&quot;http://schemas.xmlsoap.org/ws/2002/12/policy&quot;&gt;
  &lt;wsp:All wsp:Usage=&quot;Required&quot;&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;============================&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;======= Published thru API Portal =======&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;======= Don&apos;t modify block starts =======&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;============================&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:ApiPortalIntegration&gt;
      &lt;L7p:ApiGroup stringValue=&quot;&quot;/&gt;
      &lt;L7p:ApiId stringValue=&quot;43e52d60-d4e3-4c15-89c1-8764de4e3106&quot;/&gt;
      &lt;L7p:PortalManagedApiFlag stringValue=&quot;L7p:ApiPortalManagedServiceAssertion&quot;/&gt;
    &lt;/L7p:ApiPortalIntegration&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;aGVyZS5jb20=&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;apiLocation&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;ZDRkM2UyNmY=&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;serviceUrl&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;JHtwb3J0YWwubWFuYWdlZC5zZXJ2aWNlLmFwaUlkfQ==&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;counterName&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
        &lt;L7p:Encapsulated&gt;
      &lt;L7p:EncapsulatedAssertionConfigGuid stringValue=&quot;172594b6-18ba-4b0c-8d61-807db457e81d&quot;/
&gt;
      &lt;L7p:Parameters mapValue=&quot;included&quot;&gt;
            &lt;L7p:entry
                                xmlns:L7p=&quot;http://www.layer7tech.com/ws/policy&quot; &gt;
        &lt;L7p:key stringValue=&quot;sslEnabled&quot;/&gt;
        &lt;L7p:value stringValue=&quot;true&quot;/&gt;
      &lt;/L7p:entry&gt;
            &lt;L7p:entry
                                xmlns:L7p=&quot;http://www.layer7tech.com/ws/policy&quot; &gt;
        &lt;L7p:key stringValue=&quot;sla&quot;/&gt;
        &lt;L7p:value stringValue=&quot;414353&quot;/&gt;
      &lt;/L7p:entry&gt;
            &lt;L7p:entry
                                xmlns:L7p=&quot;http://www.layer7tech.com/ws/policy&quot; &gt;
        &lt;L7p:key stringValue=&quot;email&quot;/&gt;
        &lt;L7p:value stringValue=&quot;Test Value 888266&quot;/&gt;
```

```
      &lt;/L7p:entry&gt;
            &lt;L7p:entry
                                xmlns:L7p=&quot;http://www.layer7tech.com/ws/policy&quot; &gt;
        &lt;L7p:key stringValue=&quot;smtpServer&quot;/&gt;
        &lt;L7p:value stringValue=&quot;Test Value 414571&quot;/&gt;
      &lt;/L7p:entry&gt;
            &lt;/L7p:Parameters&gt;
    &lt;/L7p:Encapsulated&gt;
        &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;============================&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;======= Don&apos;t modify block ends =======&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;============================&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
  &lt;/wsp:All&gt;
&lt;/wsp:Policy&gt;
                            </l7:Resource>
                        </l7:ResourceSet>
                    </l7:Resources>
                </l7:Policy>
            </l7:Resource>
        </l7:Item>
        <l7:Item
            xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
            <l7:Name>Demo Api 294ece4a - updated</l7:Name>
            <l7:Id>144251a872943a3a8be915f4eaf1e69f</l7:Id>
            <l7:Type>SERVICE</l7:Type>
            <l7:Resource>
                <l7:Service id="144251a872943a3a8be915f4eaf1e69f"
                    xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
                    <l7:ServiceDetail folderId="ddb84c6f397d7dbd3cca71d3043f019c"
 id="144251a872943a3a8be915f4eaf1e69f">
                        <l7:Name>Demo Api 294ece4a - updated</l7:Name>
                        <l7:Enabled>true</l7:Enabled>
                        <l7:ServiceMappings>
                            <l7:HttpMapping>
                                <l7:UrlPattern>/d4d3e26f*</l7:UrlPattern>
                                <l7:Verbs>
                                    <l7:Verb>GET</l7:Verb>
                                    <l7:Verb>POST</l7:Verb>
                                    <l7:Verb>PUT</l7:Verb>
                                    <l7:Verb>DELETE</l7:Verb>
                                    <l7:Verb>OPTIONS</l7:Verb>
                                    <l7:Verb>PATCH</l7:Verb>
                                    <l7:Verb>HEAD</l7:Verb>
                                </l7:Verbs>
                            </l7:HttpMapping>
                        </l7:ServiceMappings>
                        <l7:Properties>
                            <l7:Property key="internal">
```

```
                                <l7:BooleanValue>false</l7:BooleanValue>
                            </l7:Property>
                            <l7:Property key="soap">
                                <l7:BooleanValue>false</l7:BooleanValue>
                            </l7:Property>
                            <l7:Property key="tracingEnabled">
                                <l7:BooleanValue>false</l7:BooleanValue>
                            </l7:Property>
                            <l7:Property key="wssProcessingEnabled">
                                <l7:BooleanValue>false</l7:BooleanValue>
                            </l7:Property>
                            <l7:Property key="property.portalID">
                                <l7:StringValue>43e52d60-d4e3-4c15-89c1-8764de4e3106</l7:StringValue>
                            </l7:Property>
                            <l7:Property key="property.internal.portalAPIEnabled">
                                <l7:StringValue>true</l7:StringValue>
                            </l7:Property>
                        </l7:Properties>
                    </l7:ServiceDetail>
                    <l7:Resources>
                        <l7:ResourceSet tag="policy">
                            <l7:Resource type="policy">&lt;?xml version=&quot;1.0&quot;
 encoding=&quot;UTF-8&quot;?&gt;
&lt;wsp:Policy
                                xmlns:L7p=&quot;http://www.layer7tech.com/ws/policy&quot;
                                xmlns:wsp=&quot;http://schemas.xmlsoap.org/ws/2002/12/policy&quot;&gt;
  &lt;wsp:All wsp:Usage=&quot;Required&quot;&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;------ Portal Created Fragment . Do not Modify ------&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;------- Encass has a route in it. ------&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;ZmFsc2U=&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;override_template_routing&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
    &lt;L7p:Include&gt;
      &lt;L7p:PolicyGuid stringValue=&quot;812ed196-c315-4e92-b630-b5c64c5c043c&quot;/&gt;
      &lt;L7p:PolicyName stringValue=&quot;Portal Service Preface&quot;/&gt;
    &lt;/L7p:Include&gt;
    &lt;L7p:Include&gt;
      &lt;L7p:PolicyGuid stringValue=&quot;43e52d60-d4e3-4c15-89c1-8764de4e3106&quot;/&gt;
    &lt;/L7p:Include&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;------ Portal Created Fragment . Do not Modify ------&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;------ This routing path will be executed if using override template
 routing ------&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;wsp:OneOrMore wsp:Usage=&quot;Required&quot;&gt;
      &lt;L7p:ComparisonAssertion&gt;
```

```
        &lt;L7p:CaseSensitive booleanValue=&quot;false&quot;/&gt;
        &lt;L7p:Expression1 stringValue=&quot;${override_template_routing}&quot;/&gt;
        &lt;L7p:Expression2 stringValue=&quot;false&quot;/&gt;
        &lt;L7p:Predicates predicates=&quot;included&quot;&gt;
          &lt;L7p:item binary=&quot;included&quot;&gt;
            &lt;L7p:CaseSensitive booleanValue=&quot;false&quot;/&gt;
            &lt;L7p:RightValue stringValue=&quot;false&quot;/&gt;
          &lt;/L7p:item&gt;
        &lt;/L7p:Predicates&gt;
      &lt;/L7p:ComparisonAssertion&gt;
      &lt;wsp:All wsp:Usage=&quot;Required&quot;&gt;
        &lt;L7p:SetVariable&gt;
          &lt;L7p:Base64Expression stringValue=&quot;JHtyZXF1ZXN0Lmh0dHAudXJpfQ==&quot;/&gt;
          &lt;L7p:VariableToSet stringValue=&quot;param.uri&quot;/&gt;
        &lt;/L7p:SetVariable&gt;
        &lt;L7p:Regex&gt;
          &lt;L7p:AutoTarget booleanValue=&quot;false&quot;/&gt;
          &lt;L7p:OtherTargetMessageVariable stringValue=&quot;param.uri&quot;/&gt;
          &lt;L7p:PatternContainsVariables booleanValue=&quot;true&quot;/&gt;
          &lt;L7p:Regex stringValue=&quot;/${serviceUrl}&quot;/&gt;
          &lt;L7p:RegexName stringValue=&quot;process uri&quot;/&gt;
          &lt;L7p:Replace booleanValue=&quot;true&quot;/&gt;
          &lt;L7p:Replacement stringValue=&quot;&quot;/&gt;
          &lt;L7p:Target target=&quot;OTHER&quot;/&gt;
        &lt;/L7p:Regex&gt;
        &lt;wsp:OneOrMore wsp:Usage=&quot;Required&quot;&gt;
          &lt;wsp:All wsp:Usage=&quot;Required&quot;&gt;
            &lt;L7p:HttpRoutingAssertion&gt;
              &lt;L7p:FailOnErrorStatus booleanValue=&quot;false&quot;/&gt;
              &lt;L7p:ProtectedServiceUrl stringValue=&quot;
${apiLocation}${param.uri}${request.url.query}&quot;/&gt;
              &lt;L7p:ProxyPassword stringValueNull=&quot;null&quot;/&gt;
              &lt;L7p:ProxyUsername stringValueNull=&quot;null&quot;/&gt;
              &lt;L7p:RequestHeaderRules httpPassthroughRuleSet=&quot;included&quot;&gt;
                &lt;L7p:Rules httpPassthroughRules=&quot;included&quot;&gt;
                  &lt;L7p:item httpPassthroughRule=&quot;included&quot;&gt;
                    &lt;L7p:Name stringValue=&quot;Cookie&quot;/&gt;
                  &lt;/L7p:item&gt;
                  &lt;L7p:item httpPassthroughRule=&quot;included&quot;&gt;
                    &lt;L7p:Name stringValue=&quot;SOAPAction&quot;/&gt;
                  &lt;/L7p:item&gt;
                &lt;/L7p:Rules&gt;
              &lt;/L7p:RequestHeaderRules&gt;
              &lt;L7p:RequestParamRules httpPassthroughRuleSet=&quot;included&quot;&gt;
                &lt;L7p:ForwardAll booleanValue=&quot;true&quot;/&gt;
                &lt;L7p:Rules httpPassthroughRules=&quot;included&quot;/&gt;
              &lt;/L7p:RequestParamRules&gt;
              &lt;L7p:ResponseHeaderRules httpPassthroughRuleSet=&quot;included&quot;&gt;
                &lt;L7p:Rules httpPassthroughRules=&quot;included&quot;&gt;
                  &lt;L7p:item httpPassthroughRule=&quot;included&quot;&gt;
                    &lt;L7p:Name stringValue=&quot;Set-Cookie&quot;/&gt;
                  &lt;/L7p:item&gt;
                &lt;/L7p:Rules&gt;
```

```
          &lt;/L7p:ResponseHeaderRules&gt;
          &lt;L7p:SamlAssertionVersion intValue=&quot;2&quot;/&gt;
        &lt;/L7p:HttpRoutingAssertion&gt;
        &lt;L7p:SetVariable&gt;
          &lt;L7p:Base64Expression stringValue=&quot;JHtyZXNwb25zZS5odHRwLnN0YXR1c30=&quot;/&gt;
          &lt;L7p:VariableToSet stringValue=&quot;portal.analytics.response.code&quot;/&gt;
        &lt;/L7p:SetVariable&gt;
      &lt;/wsp:All&gt;
      &lt;wsp:All wsp:Usage=&quot;Required&quot;&gt;
        &lt;L7p:SetVariable&gt;
          &lt;L7p:Base64Expression stringValue=&quot;JHtyZXNwb25zZS5odHRwLnN0YXR1c30=&quot;/&gt;
          &lt;L7p:VariableToSet stringValue=&quot;portal.analytics.response.code&quot;/&gt;
        &lt;/L7p:SetVariable&gt;
        &lt;L7p:SetVariable&gt;
          &lt;L7p:Base64Expression stringValue=&quot;VW5hYmxlIHRvIHJvdXRlIHRvIEFQSS4=&quot;/&gt;
          &lt;L7p:VariableToSet stringValue=&quot;errorMsg&quot;/&gt;
        &lt;/L7p:SetVariable&gt;
        &lt;wsp:OneOrMore wsp:Usage=&quot;Required&quot;&gt;
          &lt;L7p:ComparisonAssertion&gt;
            &lt;L7p:CaseSensitive booleanValue=&quot;false&quot;/&gt;
            &lt;L7p:Expression1 stringValue=&quot;${portal.analytics.response.code}&quot;/&gt;
            &lt;L7p:ExpressionIsVariable booleanValue=&quot;false&quot;/&gt;
            &lt;L7p:Operator operatorNull=&quot;null&quot;/&gt;
            &lt;L7p:Predicates predicates=&quot;included&quot;&gt;
              &lt;L7p:item dataType=&quot;included&quot;&gt;
                &lt;L7p:Type variableDataType=&quot;string&quot;/&gt;
              &lt;/L7p:item&gt;
              &lt;L7p:item stringLength=&quot;included&quot;&gt;
                &lt;L7p:Max intValue=&quot;-1&quot;/&gt;
                &lt;L7p:Min intValue=&quot;1&quot;/&gt;
              &lt;/L7p:item&gt;
            &lt;/L7p:Predicates&gt;
          &lt;/L7p:ComparisonAssertion&gt;
          &lt;L7p:SetVariable&gt;
            &lt;L7p:Base64Expression stringValue=&quot;NDA4&quot;/&gt;
            &lt;L7p:VariableToSet stringValue=&quot;portal.analytics.response.code&quot;/&gt;
          &lt;/L7p:SetVariable&gt;
        &lt;/wsp:OneOrMore&gt;
        &lt;L7p:FalseAssertion/&gt;
      &lt;/wsp:All&gt;
    &lt;/wsp:OneOrMore&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;JHtyZXF1ZXN0LnJvdXRpbmdUb3RhbFRpbWV9&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;portal.analytics.routingTotalTime&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
    &lt;L7p:ExportVariables&gt;
      &lt;L7p:ExportedVars stringArrayValue=&quot;included&quot;&gt;
        &lt;L7p:item stringValue=&quot;portal.analytics.response.code&quot;/&gt;
        &lt;L7p:item stringValue=&quot;portal.analytics.routingTotalTime&quot;/&gt;
      &lt;/L7p:ExportedVars&gt;
    &lt;/L7p:ExportVariables&gt;
  &lt;/wsp:All&gt;
&lt;/wsp:OneOrMore&gt;
```

```
  &lt;/wsp:All&gt;
&lt;/wsp:Policy&gt;
                            </l7:Resource>
                        </l7:ResourceSet>
                    </l7:Resources>
                </l7:Service>
            </l7:Resource>
        </l7:Item>
    </l7:References>
    <l7:Mappings>
        <l7:Mapping action="NewOrUpdate" srcId="43e52d60d4e34c1589c18764de4e3106" type="POLICY"
            xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management"/>
            <l7:Mapping action="NewOrUpdate" srcId="144251a872943a3a8be915f4eaf1e69f" type="SERVICE"
                xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management"/>
            </l7:Mappings>
            </l7:Bundle>
```

Create API on Proxy Using API Bundle

The bundle in the previous example can be used to create an API on the proxy via the REST management API (RESTman), and contains all the runtime metadata for the API such as its:

- Name
- State
- Policy template(s)
- Policy template(s) values
- Custom field(s)
- Custom field(s) values

To create the API on the proxy, using the bundle returned from the previous call:

```
curl -X PUT -H 'Content-Type: application/xml' \ -u {username}:{password} -d @api-bundle.xml \ https://
{proxyHost}/restman/1.0/bundle
```

Example request

```
curl -X PUT -H 'Content-Type: application/xml' \ -u username:password -d @api-bundle.xml \ https://atenant-
ssg.dev.ca.com:8443/restman/1.0/bundle
```

Example response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<l7:Item xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
    <l7:Name>Bundle mappings</l7:Name>
    <l7:Type>BUNDLE MAPPINGS</l7:Type>
    <l7:TimeStamp>2017-11-30T22:58:31.116Z</l7:TimeStamp>
    <l7:Link rel="self" uri="https://otk:8447/restman/1.0/bundle"/>
    <l7:Resource>
        <l7:Mappings>
            <l7:Mapping action="NewOrUpdate" actionTaken="CreatedNew" srcId="43e52d60d4e34c1589c18764de4e3106"
 targetId="43e52d60d4e34c1589c18764de4e3106" targetUri="https://otk:8447/restman/1.0/
policies/43e52d60d4e34c1589c18764de4e3106" type="POLICY"/>
            <l7:Mapping action="NewOrUpdate" actionTaken="CreatedNew" srcId="144251a872943a3a8be915f4eaf1e69f"
 targetId="144251a872943a3a8be915f4eaf1e69f" targetUri="https://otk:8447/restman/1.0/
services/144251a872943a3a8be915f4eaf1e69f" type="SERVICE"/>
        </l7:Mappings>
    </l7:Resource>
```

```
</l7:Item>
```

Now that the API has been deployed to the proxy, we can update its deployment to denote to other users it has been deployed. Update the deployment using the following command:

```
curl -X PUT -H 'Authorization: Bearer {token}' -H 'Content-Type:application/json;charset=utf-8' \ https://
{portalApiHost}/{tenantId}/deployments/1.0/apis/{apiUuid}/proxies/{proxyUuid} \ -d '{"status": "DEPLOYED"}'
```

Example request

```
curl -X PUT -H 'Authorization: Bearer 14f1f709-1b19-4c29-a0fc-c93cdc683159' -H 'Content-Type:application/
json;charset=utf-8' \ https://apim-ssg-apim-uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/
apis/43e52d60-d4e3-4c15-89c1-8764de4e3106/proxies/639536fc-230d-434e-8b88-3e13d5069c34 \ -d '{"status":
 "DEPLOYED", "message": "API deployment successful"}'
```

Example response

```
{
    "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
    "proxyUuid": "639536fc-230d-434e-8b88-3e13d5069c34",
    "proxyName": "PROD",
    "lastTimeDeployed": 1512084419249,
    "status": "DEPLOYED",
    "message": "Deployment successful"
}
```

Retrieve the API deployments to see that the deployment has been updated accordingly:

```
[
    {
        "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
        "proxyUuid": "8f6bc46c-a131-4388-b654-1e2b599b0ee9",
        "proxyName": "DEV",
        "lastTimeDeployed": 1512084601218,
        "status": "DEPLOYED"
    },
    {
        "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
        "proxyUuid": "639536fc-230d-434e-8b88-3e13d5069c34",
        "proxyName": "PROD",
        "lastTimeDeployed": 1512084419249,
        "status": "DEPLOYED",
        "message": "Deployment successful"
    }
]
```

Redeploy API

Redeploy an API if any of the API properties that impact its deployment to a Proxy are changed. Examples of the API properties:

- Name
- State
- Policy template(s)
- Policy template(s) values
- Custom field(s)
- Custom field(s) values

The API deployment status is set back to PENDING_DEPLOYMENT to indicate that API must be redeployed.

To redeploy an API for a proxy configured as scripted, repeat the previous API calls to:

1. Retrieve the API bundle.
2. Update the API on the proxy by making the same RESTman call using the updated API bundle.

Delete an API Deployment

Because API deployments and re-deployments to a proxy are done manually, undeploying an API is done manually as well.

Follow these steps:

1. Retrieve a delete API bundle using the following command:
```
curl -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/api-management/1.0/apis/
{apiUuid}/bundle
```
Example request
```
curl -H 'Authorization: Bearer 14f1f709-1b19-4c29-a0fc-c93cdc683159' \ https://apim-ssg-apim-uswest-
prod.app.prod.w2.dev.ca.com:443/atenant/api-management/1.0/apis/43e52d60-d4e3-4c15-89c1-8764de4e3106/
bundle?type=delete \ -o api-bundle.xml
```
Example response, which is saved to api-bundle.xml:
```
<l7:Bundle
    xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
    <l7:Mappings>
        <l7:Mapping action="Delete" srcId="144251a872943a3a8be915f4eaf1e69f" type="SERVICE"
            xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
        </l7:Mapping>
        <l7:Mapping action="Delete" srcId="43e52d60d4e34c1589c18764de4e3106" type="POLICY"
            xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
        </l7:Mapping>
    </l7:Mappings>
</l7:Bundle>
```
2. Delete the API on the proxy using the bundle:
```
curl -X PUT -H 'Content-Type: application/xml' \ -u {username}:{password} -d @api-bundle.xml \ https://
{proxyHost}/restman/1.0/bundle
```
Example request
```
curl -X PUT -H 'Content-Type: application/xml' \-u username:password -d @api-bundle.xml \https://atenant-
ssg.dev.ca.com:8443/restman/1.0/bundle
```
Example response
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<l7:Item xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
    <l7:Name>Bundle mappings</l7:Name>
    <l7:Type>BUNDLE MAPPINGS</l7:Type>
    <l7:TimeStamp>2017-11-30T23:43:50.634Z</l7:TimeStamp>
    <l7:Link rel="self" uri="https://otk:8447/restman/1.0/bundle"/>
    <l7:Resource>
        <l7:Mappings>
            <l7:Mapping action="Delete" actionTaken="Deleted" srcId="144251a872943a3a8be915f4eaf1e69f"
 targetId="144251a872943a3a8be915f4eaf1e69f" targetUri="https://otk:8447/restman/1.0/
services/144251a872943a3a8be915f4eaf1e69f" type="SERVICE"/>
            <l7:Mapping action="Delete" actionTaken="Deleted" srcId="43e52d60d4e34c1589c18764de4e3106"
 targetId="43e52d60d4e34c1589c18764de4e3106" targetUri="https://otk:8447/restman/1.0/
policies/43e52d60d4e34c1589c18764de4e3106" type="POLICY"/>
        </l7:Mappings>
```

```
        </l7:Resource>
    </l7:Item>
```

3. With the API deleted from the proxy, delete the deployment:

```
curl -X DELETE -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/deployments/1.0/apis/
{apiUuid}/proxies/{proxyUuid}
```

Example request

```
curl -X DELETE -H 'Authorization: Bearer 14f1f709-1b19-4c29-a0fc-c93cdc683159' \ https://apim-ssg-apim-
uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/apis/43e52d60-d4e3-4c15-89c1-8764de4e3106/
proxies/639536fc-230d-434e-8b88-3e13d5069c34
```

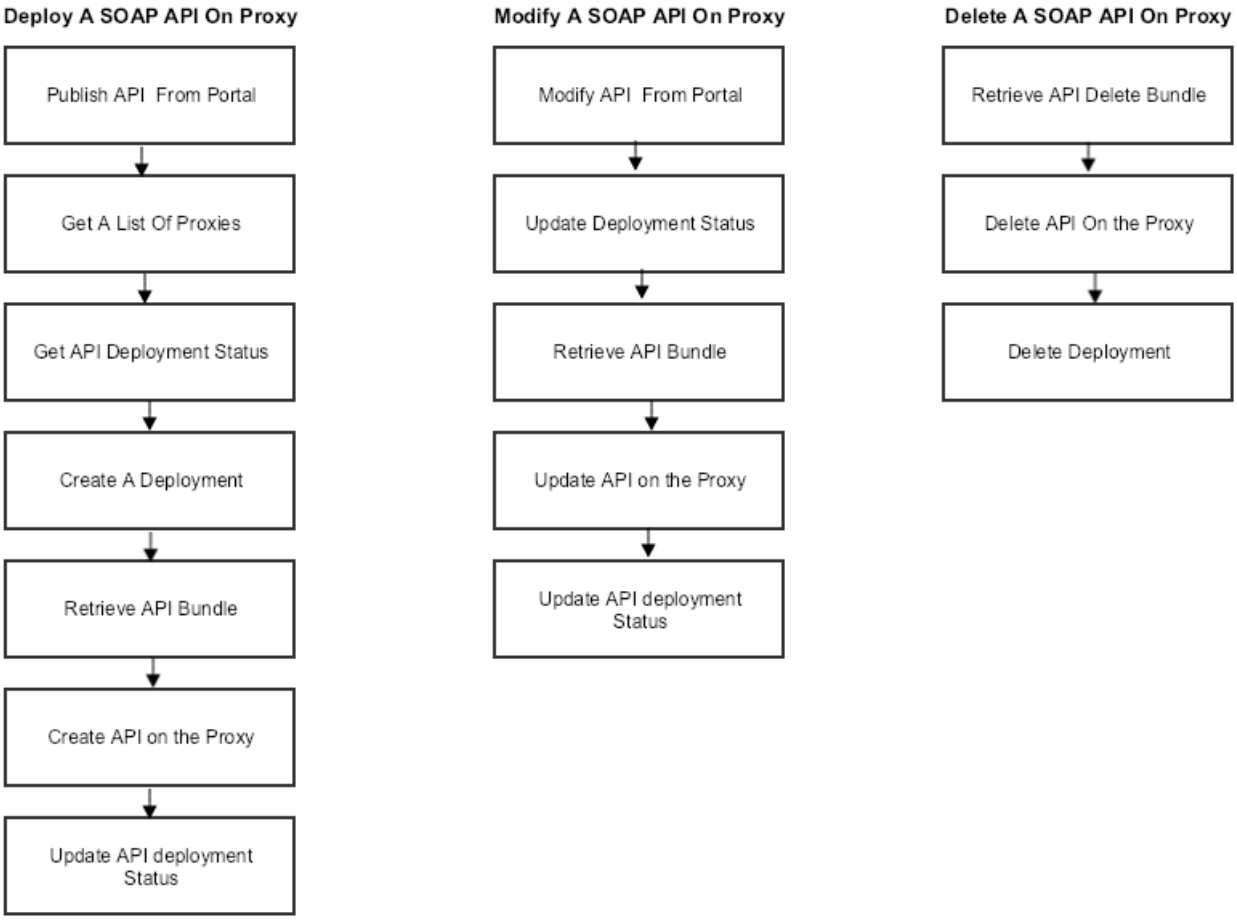
Example response shows that deployment has been deleted:

```
[
    {
        "apiUuid": "43e52d60-d4e3-4c15-89c1-8764de4e3106",
        "proxyUuid": "8f6bc46c-a131-4388-b654-1e2b599b0ee9",
        "proxyName": "DEV",
        "lastTimeDeployed": 1512001260997,
        "status": "DEPLOYED"
    }
]
```

SOAP API Scripted Deployment

This topic includes instructions for deploying, modifying, and undeploying a SOAP API scripted deployment.

The workflows are as follows:



Deploy a SOAP API with Scripted Deployment

Follow these steps:

1. Get a list of proxies:

```
curl -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/deployments/1.0/proxies
```

Example request

```
curl -H "Authorization:Bearer 5d9646b0-6290-412c-98c3-82cba4c88fce" --request GET "https://apim-ssg-soapsupport.app.dev1.w2.saasqa.ca.com:443/tenantsoapsupport/deployments/1.0/proxies"
```

Example response

```
[
   {
      "uuid":"845b69bc-0aa6-425c-91d3-99a37bca0221",
      "name":"UAT",
      "enrollmentStatus":"ACTIVE",
      "deploymentType":"MANUAL"
   }
]
```

2. Get API deployment status:

```
curl -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/deployments/1.0/apis/{apiUuid}/proxies
```

Example request

```
curl -H 'Authorization: Bearer 5d9646b0-6290-412c-98c3-82cba4c88fce' \
https://apim-ssg-soapsupport.app.dev1.w2.saasqa.ca.com:443/tenantsoapsupport/deployments/1.0/apis/
d77f1e16-c899-4556-8a16-09c283cb8f8d/proxies
```

Example response

```
[]
```

3. Create a deployment:

```
curl -X POST -H 'Authorization: Bearer {token}' -H 'Content-Type:application/json;charset=utf-8' \
https://{portalApiHost}/{tenantId}/deployments/1.0/apis/{apiUuid}/proxies \
-d '{"proxyUuid": "{proxyUuid}"}'
```

Example request

```
curl -X POST -H 'Authorization: Bearer 5d9646b0-6290-412c-98c3-82cba4c88fce' -H 'Content-Type:application/
json;charset=utf-8' \
https://apim-ssg-soapsupport.app.dev1.w2.saasqa.ca.com:443/tenantsoapsupport/deployments/1.0/apis/
d77f1e16-c899-4556-8a16-09c283cb8f8d/proxies \
-d '{"proxyUuid": "845b69bc-0aa6-425c-91d3-99a37bca0221"}'
```

Example response

```
{
   "apiUuid":"d77f1e16-c899-4556-8a16-09c283cb8f8d",
   "proxyUuid":"845b69bc-0aa6-425c-91d3-99a37bca0221",
   "proxyName":"UAT",
   "lastTimeDeployed":0,
   "status":"PENDING_DEPLOYMENT"
}
```

4. Retrieve API bundle:

```
curl -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/api-management/1.0/apis/
{apiUuid}/bundle
```

Example request

```
curl -H 'Authorization: Bearer 5d9646b0-6290-412c-98c3-82cba4c88fce' \
https://apim-ssg-soapsupport.app.dev1.w2.saasqa.ca.com:443/tenantsoapsupport/api-management/1.0/apis/
d77f1e16-c899-4556-8a16-09c283cb8f8d/bundle \
-o soap-api-bundle.xml
```

Content of soap-api-bundle.xml

```
<l7:Bundle xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
  <l7:References>
    <l7:Item xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
      <l7:Name>soapapidemo-fragment</l7:Name>
      <l7:Id>d77f1e16c89945568a1609c283cb8f8d</l7:Id>
      <l7:Type>POLICY</l7:Type>
      <l7:Resource>
        <l7:Policy guid="d77f1e16-c899-4556-8a16-09c283cb8f8d" id="d77f1e16c89945568a1609c283cb8f8d"
 version="0">
          <l7:PolicyDetail folderId="ddb84c6f397d7dbd3cca71d3043f019c" guid="d77f1e16-
c899-4556-8a16-09c283cb8f8d" id="d77f1e16c89945568a1609c283cb8f8d">
            <l7:Name>soapapidemo-fragment</l7:Name>
            <l7:PolicyType>Include</l7:PolicyType>
            <l7:Properties>
              <l7:Property key="revision">
                <l7:LongValue>1</l7:LongValue>
```

```
              </l7:Property>
              <l7:Property key="soap">
                <l7:BooleanValue>true</l7:BooleanValue>
              </l7:Property>
            </l7:Properties>
          </l7:PolicyDetail>
          <l7:Resources>
            <l7:ResourceSet tag="policy">
              <l7:Resource type="policy">&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;
&lt;wsp:Policy xmlns:L7p=&quot;http://www.layer7tech.com/ws/policy&quot; xmlns:wsp=&quot;http://
schemas.xmlsoap.org/ws/2002/12/policy&quot;&gt;
  &lt;wsp:All wsp:Usage=&quot;Required&quot;&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;============================&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;======= Published thru API Portal =======&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;======= Don&apos;t modify block starts =======&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;============================&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:ApiPortalIntegration&gt;
      &lt;L7p:ApiGroup stringValue=&quot;&quot;/&gt;
      &lt;L7p:ApiId stringValue=&quot;d77f1e16-c899-4556-8a16-09c283cb8f8d&quot;/&gt;
      &lt;L7p:PortalManagedApiFlag stringValue=&quot;L7p:ApiPortalManagedServiceAssertion&quot;/&gt;
    &lt;/L7p:ApiPortalIntegration&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;aHR0cDovL2xvY2FsaG9zdDo4MDgwL2VjaG8=&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;apiLocation&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;c29hcGFwaWRlbW8=&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;serviceUrl&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;JHtwb3J0YWwubWFuYWdlZC5zZXJ2aWNlLmFwaUlkfQ==&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;counterName&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
        &lt;L7p:Encapsulated&gt;
      &lt;L7p:EncapsulatedAssertionConfigGuid stringValue=&quot;72093738-871a-45bd-b114-
ad3a61893ac0&quot;/&gt;
      &lt;L7p:Parameters mapValue=&quot;included&quot;&gt;
            &lt;/L7p:Parameters&gt;
    &lt;/L7p:Encapsulated&gt;
        &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;============================&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;======= Don&apos;t modify block ends =======&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
```

```
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;============================&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
          &lt;L7p:HttpRoutingAssertion&gt;
        &lt;L7p:FailOnErrorStatus booleanValue=&quot;false&quot;/&gt;
        &lt;L7p:ProtectedServiceUrl stringValue=&quot;
${apiLocation}${param.uri}${request.url.query}&quot;/&gt; &lt;!-- &lt;soap:address location=&quot;http://
www.webservicex.net/stockquote.asmx&quot; /&gt; --&gt;
        &lt;L7p:ProxyPassword stringValueNull=&quot;null&quot;/&gt;
        &lt;L7p:ProxyUsername stringValueNull=&quot;null&quot;/&gt;
        &lt;L7p:RequestHeaderRules httpPassthroughRuleSet=&quot;included&quot;&gt;
          &lt;L7p:Rules httpPassthroughRules=&quot;included&quot;&gt;
            &lt;L7p:item httpPassthroughRule=&quot;included&quot;&gt;
              &lt;L7p:Name stringValue=&quot;Cookie&quot;/&gt;
            &lt;/L7p:item&gt;
            &lt;L7p:item httpPassthroughRule=&quot;included&quot;&gt;
              &lt;L7p:Name stringValue=&quot;SOAPAction&quot;/&gt;
            &lt;/L7p:item&gt;
          &lt;/L7p:Rules&gt;
        &lt;/L7p:RequestHeaderRules&gt;
        &lt;L7p:RequestParamRules httpPassthroughRuleSet=&quot;included&quot;&gt;
          &lt;L7p:ForwardAll booleanValue=&quot;true&quot;/&gt;
          &lt;L7p:Rules httpPassthroughRules=&quot;included&quot;/&gt;
        &lt;/L7p:RequestParamRules&gt;
        &lt;L7p:ResponseHeaderRules httpPassthroughRuleSet=&quot;included&quot;&gt;
          &lt;L7p:Rules httpPassthroughRules=&quot;included&quot;&gt;
            &lt;L7p:item httpPassthroughRule=&quot;included&quot;&gt;
              &lt;L7p:Name stringValue=&quot;Set-Cookie&quot;/&gt;
            &lt;/L7p:item&gt;
          &lt;/L7p:Rules&gt;
        &lt;/L7p:ResponseHeaderRules&gt;
        &lt;L7p:SamlAssertionVersion intValue=&quot;2&quot;/&gt;
      &lt;/L7p:HttpRoutingAssertion&gt;
      &lt;/wsp:All&gt;
&lt;/wsp:Policy&gt;</l7:Resource>
            </l7:ResourceSet>
          </l7:Resources>
        </l7:Policy>
      </l7:Resource>
    </l7:Item>
    <l7:Item xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
      <l7:Name>soapapidemo</l7:Name>
      <l7:Id>b2286ad6a8243794af20976095151300</l7:Id>
      <l7:Type>SERVICE</l7:Type>
      <l7:Resource>
        <l7:Service id="b2286ad6a8243794af20976095151300" xmlns:l7="http://ns.l7tech.com/2010/04/gateway-
management">
          <l7:ServiceDetail folderId="ddb84c6f397d7dbd3cca71d3043f019c"
 id="b2286ad6a8243794af20976095151300">
            <l7:Name>soapapidemo</l7:Name>
            <l7:Enabled>true</l7:Enabled>
            <l7:ServiceMappings>
              <l7:HttpMapping>
```

```
                <l7:UrlPattern>/soapapidemo*</l7:UrlPattern>
                <l7:Verbs>
                                    <l7:Verb>POST</l7:Verb>
                              </l7:Verbs>
              </l7:HttpMapping>
            </l7:ServiceMappings>
            <l7:Properties>
              <l7:Property key="internal">
                <l7:BooleanValue>false</l7:BooleanValue>
              </l7:Property>
              <l7:Property key="soap">
                <l7:BooleanValue>true</l7:BooleanValue>
              </l7:Property>
              <l7:Property key="tracingEnabled">
                <l7:BooleanValue>false</l7:BooleanValue>
              </l7:Property>
              <l7:Property key="wssProcessingEnabled">
                <l7:BooleanValue>false</l7:BooleanValue>
              </l7:Property>
              <l7:Property key="property.portalID">
                <l7:StringValue>d77f1e16-c899-4556-8a16-09c283cb8f8d</l7:StringValue>
              </l7:Property>
              <l7:Property key="property.internal.portalAPIEnabled">
                <l7:StringValue>true</l7:StringValue>
              </l7:Property>
              <l7:Property key="property.portalModifyTS">
                <l7:StringValue>1556848228104</l7:StringValue>
              </l7:Property>
                          </l7:Properties>
          </l7:ServiceDetail>
          <l7:Resources>
            <l7:ResourceSet tag="policy">
              <l7:Resource type="policy">&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;
&lt;wsp:Policy xmlns:L7p=&quot;http://www.layer7tech.com/ws/policy&quot; xmlns:wsp=&quot;http://
schemas.xmlsoap.org/ws/2002/12/policy&quot;&gt;
  &lt;wsp:All wsp:Usage=&quot;Required&quot;&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;------ Portal Created Fragment . Do not Modify ------&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;------- Encass has a route in it. ------&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;ZmFsc2U=&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;override_template_routing&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
    &lt;L7p:Include&gt;
      &lt;L7p:PolicyGuid stringValue=&quot;812ed196-c315-4e92-b630-b5c64c5c043c&quot;/&gt;
      &lt;L7p:PolicyName stringValue=&quot;Portal Service Preface&quot;/&gt;
    &lt;/L7p:Include&gt;
    &lt;L7p:Include&gt;
      &lt;L7p:PolicyGuid stringValue=&quot;d77f1e16-c899-4556-8a16-09c283cb8f8d&quot;/&gt;
    &lt;/L7p:Include&gt;
```

```
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;------ Portal Created Fragment . Do not Modify ------&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;------ This routing path will be executed if using override
 template routing ------&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;wsp:OneOrMore wsp:Usage=&quot;Required&quot;&gt;
      &lt;L7p:ComparisonAssertion&gt;
        &lt;L7p:CaseSensitive booleanValue=&quot;false&quot;/&gt;
        &lt;L7p:Expression1 stringValue=&quot;${override_template_routing}&quot;/&gt;
        &lt;L7p:Expression2 stringValue=&quot;false&quot;/&gt;
        &lt;L7p:Predicates predicates=&quot;included&quot;&gt;
          &lt;L7p:item binary=&quot;included&quot;&gt;
            &lt;L7p:CaseSensitive booleanValue=&quot;false&quot;/&gt;
            &lt;L7p:RightValue stringValue=&quot;false&quot;/&gt;
          &lt;/L7p:item&gt;
        &lt;/L7p:Predicates&gt;
      &lt;/L7p:ComparisonAssertion&gt;
      &lt;wsp:All wsp:Usage=&quot;Required&quot;&gt;
        &lt;L7p:SetVariable&gt;
          &lt;L7p:Base64Expression stringValue=&quot;JHtyZXF1ZXN0Lmh0dHAudXJpfQ==&quot;/&gt;
          &lt;L7p:VariableToSet stringValue=&quot;param.uri&quot;/&gt;
        &lt;/L7p:SetVariable&gt;
        &lt;L7p:Regex&gt;
          &lt;L7p:AutoTarget booleanValue=&quot;false&quot;/&gt;
          &lt;L7p:OtherTargetMessageVariable stringValue=&quot;param.uri&quot;/&gt;
          &lt;L7p:PatternContainsVariables booleanValue=&quot;true&quot;/&gt;
          &lt;L7p:Regex stringValue=&quot;/${serviceUrl}&quot;/&gt;
          &lt;L7p:RegexName stringValue=&quot;process uri&quot;/&gt;
          &lt;L7p:Replace booleanValue=&quot;true&quot;/&gt;
          &lt;L7p:Replacement stringValue=&quot;&quot;/&gt;
          &lt;L7p:Target target=&quot;OTHER&quot;/&gt;
        &lt;/L7p:Regex&gt;
        &lt;wsp:OneOrMore wsp:Usage=&quot;Required&quot;&gt;
          &lt;wsp:All wsp:Usage=&quot;Required&quot;&gt;
            &lt;L7p:HttpRoutingAssertion&gt;
              &lt;L7p:FailOnErrorStatus booleanValue=&quot;false&quot;/&gt;
              &lt;L7p:ProtectedServiceUrl stringValue=&quot;
${apiLocation}${param.uri}${request.url.query}&quot;/&gt;
              &lt;L7p:ProxyPassword stringValueNull=&quot;null&quot;/&gt;
              &lt;L7p:ProxyUsername stringValueNull=&quot;null&quot;/&gt;
              &lt;L7p:RequestHeaderRules httpPassthroughRuleSet=&quot;included&quot;&gt;
                &lt;L7p:Rules httpPassthroughRules=&quot;included&quot;&gt;
                  &lt;L7p:item httpPassthroughRule=&quot;included&quot;&gt;
                    &lt;L7p:Name stringValue=&quot;Cookie&quot;/&gt;
                  &lt;/L7p:item&gt;
                  &lt;L7p:item httpPassthroughRule=&quot;included&quot;&gt;
                    &lt;L7p:Name stringValue=&quot;SOAPAction&quot;/&gt;
                  &lt;/L7p:item&gt;
                &lt;/L7p:Rules&gt;
              &lt;/L7p:RequestHeaderRules&gt;
              &lt;L7p:RequestParamRules httpPassthroughRuleSet=&quot;included&quot;&gt;
```

```
        &lt;L7p:ForwardAll booleanValue=&quot;true&quot;/&gt;
        &lt;L7p:Rules httpPassthroughRules=&quot;included&quot;/&gt;
      &lt;/L7p:RequestParamRules&gt;
      &lt;L7p:ResponseHeaderRules httpPassthroughRuleSet=&quot;included&quot;&gt;
        &lt;L7p:Rules httpPassthroughRules=&quot;included&quot;&gt;
          &lt;L7p:item httpPassthroughRule=&quot;included&quot;&gt;
            &lt;L7p:Name stringValue=&quot;Set-Cookie&quot;/&gt;
          &lt;/L7p:item&gt;
        &lt;/L7p:Rules&gt;
      &lt;/L7p:ResponseHeaderRules&gt;
      &lt;L7p:SamlAssertionVersion intValue=&quot;2&quot;/&gt;
    &lt;/L7p:HttpRoutingAssertion&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;JHtyZXNwb25zZS5odHRwLnN0YXR1c30=&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;portal.analytics.response.code&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
  &lt;/wsp:All&gt;
  &lt;wsp:All wsp:Usage=&quot;Required&quot;&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;JHtyZXNwb25zZS5odHRwLnN0YXR1c30=&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;portal.analytics.response.code&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;VW5hYmxlIHRvIHJvdXRlIHRvIEFQSS4=&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;errorMsg&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
    &lt;wsp:OneOrMore wsp:Usage=&quot;Required&quot;&gt;
      &lt;L7p:ComparisonAssertion&gt;
        &lt;L7p:CaseSensitive booleanValue=&quot;false&quot;/&gt;
        &lt;L7p:Expression1 stringValue=&quot;${portal.analytics.response.code}&quot;/&gt;
        &lt;L7p:ExpressionIsVariable booleanValue=&quot;false&quot;/&gt;
        &lt;L7p:Operator operatorNull=&quot;null&quot;/&gt;
        &lt;L7p:Predicates predicates=&quot;included&quot;&gt;
          &lt;L7p:item dataType=&quot;included&quot;&gt;
            &lt;L7p:Type variableDataType=&quot;string&quot;/&gt;
          &lt;/L7p:item&gt;
          &lt;L7p:item stringLength=&quot;included&quot;&gt;
            &lt;L7p:Max intValue=&quot;-1&quot;/&gt;
            &lt;L7p:Min intValue=&quot;1&quot;/&gt;
          &lt;/L7p:item&gt;
        &lt;/L7p:Predicates&gt;
      &lt;/L7p:ComparisonAssertion&gt;
      &lt;L7p:SetVariable&gt;
        &lt;L7p:Base64Expression stringValue=&quot;NDA4&quot;/&gt;
        &lt;L7p:VariableToSet stringValue=&quot;portal.analytics.response.code&quot;/&gt;
      &lt;/L7p:SetVariable&gt;
    &lt;/wsp:OneOrMore&gt;
    &lt;L7p:FalseAssertion/&gt;
  &lt;/wsp:All&gt;
&lt;/wsp:OneOrMore&gt;
&lt;L7p:SetVariable&gt;
  &lt;L7p:Base64Expression stringValue=&quot;JHtyZXF1ZXN0LnJvdXRpbmdUb3RhbFRpbWV9&quot;/&gt;
  &lt;L7p:VariableToSet stringValue=&quot;portal.analytics.routingTotalTime&quot;/&gt;
```

```
        &lt;/L7p:SetVariable&gt;
        &lt;L7p:ExportVariables&gt;
          &lt;L7p:ExportedVars stringArrayValue=&quot;included&quot;&gt;
            &lt;L7p:item stringValue=&quot;portal.analytics.response.code&quot;/&gt;
            &lt;L7p:item stringValue=&quot;portal.analytics.routingTotalTime&quot;/&gt;
          &lt;/L7p:ExportedVars&gt;
        &lt;/L7p:ExportVariables&gt;
      &lt;/wsp:All&gt;
    &lt;/wsp:OneOrMore&gt;
  &lt;/wsp:All&gt;
&lt;/wsp:Policy&gt;</l7:Resource>
            </l7:ResourceSet>
                          <l7:ResourceSet tag="wsdl">
                <l7:Resource type="wsdl">&lt;wsdl:definitions xmlns:wsdl=&quot;http://schemas.xmlsoap.org/
wsdl/&quot; xmlns:ns1=&quot;http://org.apache.axis2/xsd&quot; xmlns:ns=&quot;http://c.b.a&quot;
 xmlns:wsaw=&quot;http://www.w3.org/2006/05/addressing/wsdl&quot; xmlns:http=&quot;http://
schemas.xmlsoap.org/wsdl/http/&quot; xmlns:xs=&quot;http://www.w3.org/2001/XMLSchema&quot;
 xmlns:soap=&quot;http://schemas.xmlsoap.org/wsdl/soap/&quot; xmlns:mime=&quot;http://
schemas.xmlsoap.org/wsdl/mime/&quot; xmlns:soap12=&quot;http://schemas.xmlsoap.org/wsdl/soap12/&quot;
 targetNamespace=&quot;http://c.b.a&quot;&gt;
    &lt;wsdl:documentation&gt;Calculator&lt;/wsdl:documentation&gt;
    &lt;wsdl:types&gt;
        &lt;xs:schema attributeFormDefault=&quot;qualified&quot; elementFormDefault=&quot;qualified&quot;
 targetNamespace=&quot;http://c.b.a&quot;&gt;
            &lt;xs:element name=&quot;add&quot;&gt;
                &lt;xs:complexType&gt;
                    &lt;xs:sequence&gt;
                        &lt;xs:element minOccurs=&quot;0&quot; name=&quot;n1&quot;
 type=&quot;xs:int&quot; /&gt;
                        &lt;xs:element minOccurs=&quot;0&quot; name=&quot;n2&quot;
 type=&quot;xs:int&quot; /&gt;
                    &lt;/xs:sequence&gt;
                &lt;/xs:complexType&gt;
            &lt;/xs:element&gt;
            &lt;xs:element name=&quot;addResponse&quot;&gt;
                &lt;xs:complexType&gt;
                    &lt;xs:sequence&gt;
                        &lt;xs:element minOccurs=&quot;0&quot; name=&quot;return&quot;
 type=&quot;xs:int&quot; /&gt;
                    &lt;/xs:sequence&gt;
                &lt;/xs:complexType&gt;
            &lt;/xs:element&gt;
        &lt;/xs:schema&gt;
    &lt;/wsdl:types&gt;
    &lt;wsdl:message name=&quot;addRequest&quot;&gt;
        &lt;wsdl:part name=&quot;parameters&quot; element=&quot;ns:add&quot; /&gt;
    &lt;/wsdl:message&gt;
    &lt;wsdl:message name=&quot;addResponse&quot;&gt;
        &lt;wsdl:part name=&quot;parameters&quot; element=&quot;ns:addResponse&quot; /&gt;
    &lt;/wsdl:message&gt;
    &lt;wsdl:portType name=&quot;CalculatorPortType&quot;&gt;
        &lt;wsdl:operation name=&quot;add&quot;&gt;
            &lt;wsdl:input message=&quot;ns:addRequest&quot; wsaw:Action=&quot;urn:add&quot; /&gt;
```

```
            &lt;wsdl:output message=&quot;ns:addResponse&quot; wsaw:Action=&quot;urn:addResponse&quot; /
&gt;
        &lt;/wsdl:operation&gt;
    &lt;/wsdl:portType&gt;
    &lt;wsdl:binding name=&quot;CalculatorSoap11Binding&quot; type=&quot;ns:CalculatorPortType&quot;&gt;
        &lt;soap:binding transport=&quot;http://schemas.xmlsoap.org/soap/http&quot;
 style=&quot;document&quot; /&gt;
        &lt;wsdl:operation name=&quot;add&quot;&gt;
            &lt;soap:operation soapAction=&quot;urn:add&quot; style=&quot;document&quot; /&gt;
            &lt;wsdl:input&gt;
                &lt;soap:body use=&quot;literal&quot; /&gt;
            &lt;/wsdl:input&gt;
            &lt;wsdl:output&gt;
                &lt;soap:body use=&quot;literal&quot; /&gt;
            &lt;/wsdl:output&gt;
        &lt;/wsdl:operation&gt;
    &lt;/wsdl:binding&gt;
    &lt;wsdl:binding name=&quot;CalculatorSoap12Binding&quot; type=&quot;ns:CalculatorPortType&quot;&gt;
        &lt;soap12:binding transport=&quot;http://schemas.xmlsoap.org/soap/http&quot;
 style=&quot;document&quot; /&gt;
        &lt;wsdl:operation name=&quot;add&quot;&gt;
            &lt;soap12:operation soapAction=&quot;urn:add&quot; style=&quot;document&quot; /&gt;
            &lt;wsdl:input&gt;
                &lt;soap12:body use=&quot;literal&quot; /&gt;
            &lt;/wsdl:input&gt;
            &lt;wsdl:output&gt;
                &lt;soap12:body use=&quot;literal&quot; /&gt;
            &lt;/wsdl:output&gt;
        &lt;/wsdl:operation&gt;
    &lt;/wsdl:binding&gt;
    &lt;wsdl:binding name=&quot;CalculatorHttpBinding&quot; type=&quot;ns:CalculatorPortType&quot;&gt;
        &lt;http:binding verb=&quot;POST&quot; /&gt;
        &lt;wsdl:operation name=&quot;add&quot;&gt;
            &lt;http:operation location=&quot;add&quot; /&gt;
            &lt;wsdl:input&gt;
                &lt;mime:content type=&quot;text/xml&quot; part=&quot;parameters&quot; /&gt;
            &lt;/wsdl:input&gt;
            &lt;wsdl:output&gt;
                &lt;mime:content type=&quot;text/xml&quot; part=&quot;parameters&quot; /&gt;
            &lt;/wsdl:output&gt;
        &lt;/wsdl:operation&gt;
    &lt;/wsdl:binding&gt;
    &lt;wsdl:service name=&quot;Calculator&quot;&gt;
        &lt;wsdl:port name=&quot;CalculatorHttpsSoap11Endpoint&quot;
 binding=&quot;ns:CalculatorSoap11Binding&quot;&gt;
            &lt;soap:address location=&quot;https://156.56.179.164:9443/services/
Calculator.CalculatorHttpsSoap11Endpoint/&quot; /&gt;
        &lt;/wsdl:port&gt;
        &lt;wsdl:port name=&quot;CalculatorHttpSoap11Endpoint&quot;
 binding=&quot;ns:CalculatorSoap11Binding&quot;&gt;
            &lt;soap:address location=&quot;http://156.56.179.164:9763/services/
Calculator.CalculatorHttpSoap11Endpoint/&quot; /&gt;
        &lt;/wsdl:port&gt;
```

```
        &lt;wsdl:port name=&quot;CalculatorHttpSoap12Endpoint&quot;
 binding=&quot;ns:CalculatorSoap12Binding&quot;&gt;
            &lt;soap12:address location=&quot;http://156.56.179.164:9763/services/
Calculator.CalculatorHttpSoap12Endpoint/&quot; /&gt;
        &lt;/wsdl:port&gt;
        &lt;wsdl:port name=&quot;CalculatorHttpsSoap12Endpoint&quot;
 binding=&quot;ns:CalculatorSoap12Binding&quot;&gt;
            &lt;soap12:address location=&quot;https://156.56.179.164:9443/services/
Calculator.CalculatorHttpsSoap12Endpoint/&quot; /&gt;
        &lt;/wsdl:port&gt;
        &lt;wsdl:port name=&quot;CalculatorHttpsEndpoint&quot;
 binding=&quot;ns:CalculatorHttpBinding&quot;&gt;
            &lt;http:address location=&quot;https://156.56.179.164:9443/services/
Calculator.CalculatorHttpsEndpoint/&quot; /&gt;
        &lt;/wsdl:port&gt;
        &lt;wsdl:port name=&quot;CalculatorHttpEndpoint&quot;
 binding=&quot;ns:CalculatorHttpBinding&quot;&gt;
            &lt;http:address location=&quot;http://156.56.179.164:9763/services/
Calculator.CalculatorHttpEndpoint/&quot; /&gt;
        &lt;/wsdl:port&gt;
    &lt;/wsdl:service&gt;
&lt;/wsdl:definitions&gt;

</l7:Resource>
              </l7:ResourceSet>
                      </l7:Resources>
        </l7:Service>
      </l7:Resource>
    </l7:Item>
  </l7:References>
  <l7:Mappings>
    <l7:Mapping action="NewOrUpdate" srcId="d77f1e16c89945568a1609c283cb8f8d" type="POLICY"
 xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management"/>
    <l7:Mapping action="NewOrUpdate" srcId="b2286ad6a8243794af20976095151300" type="SERVICE"
 xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management"/>
  </l7:Mappings>
</l7:Bundle>
```

5. Create API on the proxy:

```
curl -X PUT -H 'Content-Type: application/xml' \
-u {username}:{password} -d @api-bundle.xml \
https://{proxyHost}/restman/1.0/bundle
```

Example request

```
curl -X PUT -H 'Content-Type: application/xml' \
-u admin:password -d @soap-api-bundle.xml \
https://ohgateway1-94.lvn.broadcom.net:8443/restman/1.0/bundle
```

Example response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<l7:Item xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
    <l7:Name>Bundle mappings</l7:Name>
    <l7:Type>BUNDLE MAPPINGS</l7:Type>
    <l7:TimeStamp>2019-05-06T11:13:39.669-07:00</l7:TimeStamp>
    <l7:Link rel="self" uri="https://ohgateway1-94.lvn.broadcom.net:8443/restman/1.0/bundle"/>
```

```
    <l7:Resource>
        <l7:Mappings>
            <l7:Mapping action="NewOrUpdate" actionTaken="CreatedNew"
 srcId="d77f1e16c89945568a1609c283cb8f8d" targetId="d77f1e16c89945568a1609c283cb8f8d" targetUri="https://
ohgateway1-94.lvn.broadcom.net:8443/restman/1.0/policies/d77f1e16c89945568a1609c283cb8f8d" type="POLICY"/>
            <l7:Mapping action="NewOrUpdate" actionTaken="CreatedNew"
 srcId="b2286ad6a8243794af20976095151300" targetId="b2286ad6a8243794af20976095151300" targetUri="https://
ohgateway1-94.lvn.broadcom.net:8443/restman/1.0/services/b2286ad6a8243794af20976095151300" type="SERVICE"/
>
        </l7:Mappings>
    </l7:Resource>
</l7:Item>
```

6. Update API deployment status:

```
curl -X PUT -H 'Authorization: Bearer {token}' -H 'Content-Type:application/json;charset=utf-8' \
https://{portalApiHost}/{tenantId}/deployments/1.0/apis/{apiUuid}/proxies/{proxyUuid} \
-d '{"status": "DEPLOYED"}'
```

Example request

```
curl -X PUT -H 'Authorization: Bearer 5d9646b0-6290-412c-98c3-82cba4c88fce' -H 'Content-Type:application/
json;charset=utf-8' \
https://apim-ssg-soapsupport.app.dev1.w2.saasqa.ca.com:443/tenantsoapsupport/deployments/1.0/apis/
d77f1e16-c899-4556-8a16-09c283cb8f8d/proxies/845b69bc-0aa6-425c-91d3-99a37bca0221 \
-d '{"status": "DEPLOYED", "message": "API deployment successful"}'
```

Example response

```
{
"apiUuid":"d77f1e16-c899-4556-8a16-09c283cb8f8d",
"proxyUuid":"845b69bc-0aa6-425c-91d3-99a37bca0221",
"proxyName":"UAT",
"lastTimeDeployed":1557166823186,
"status":"DEPLOYED",
"message":"API deployment successful"
}
```

Modify a Scripted Deployment

Follow these steps:

1. Update deployment status:

```
curl -X PUT -H 'Authorization: Bearer {token}' -H 'Content-Type:application/json;charset=utf-8' \
https://{portalApiHost}/{tenantId}/deployments/1.0/apis/{apiUuid}/proxies/{proxyUuid} \
-d '{"status": "PENDING_DEPLOYMENT"}'
```

Example request

```
curl -X PUT -H 'Authorization: Bearer aa221b8d-5101-428c-b6cc-c9a2fe94f2b0' -H 'Content-Type:application/
json;charset=utf-8' \
https://apim-ssg-soapsupport.app.dev1.w2.saasqa.ca.com:443/tenantsoapsupport/deployments/1.0/apis/
d77f1e16-c899-4556-8a16-09c283cb8f8d/proxies/845b69bc-0aa6-425c-91d3-99a37bca0221 \
-d '{"status": "PENDING_DEPLOYMENT"}'
```

Example response

```
{
   "apiUuid":"d77f1e16-c899-4556-8a16-09c283cb8f8d",
   "proxyUuid":"845b69bc-0aa6-425c-91d3-99a37bca0221",
   "proxyName":"UAT",
   "lastTimeDeployed":1557252051085,
```

```
    "status":"PENDING_DEPLOYMENT"
}
```

2. Retrieve the API bundle:

```
curl -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/api-management/1.0/apis/
{apiUuid}/bundle
```

Example request

```
curl -k -H 'Authorization: Bearer 5d9646b0-6290-412c-98c3-82cba4c88fce' https://apim-ssg-
soapsupport.app.dev1.w2.saasqa.ca.com:443/tenantsoapsupport/api-management/1.0/apis/d77f1e16-
c899-4556-8a16-09c283cb8f8d/bundle -o soap-api-bundle_updated.xml
```

Content of soap-api-bundle_updated.xml

```
<l7:Bundle xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
  <l7:References>
    <l7:Item xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
      <l7:Name>soapapidemo-edit-fragment</l7:Name>
      <l7:Id>d77f1e16c89945568a1609c283cb8f8d</l7:Id>
      <l7:Type>POLICY</l7:Type>
      <l7:Resource>
        <l7:Policy guid="d77f1e16-c899-4556-8a16-09c283cb8f8d" id="d77f1e16c89945568a1609c283cb8f8d"
 version="0">
          <l7:PolicyDetail folderId="ddb84c6f397d7dbd3cca71d3043f019c" guid="d77f1e16-
c899-4556-8a16-09c283cb8f8d" id="d77f1e16c89945568a1609c283cb8f8d">
            <l7:Name>soapapidemo-edit-fragment</l7:Name>
            <l7:PolicyType>Include</l7:PolicyType>
            <l7:Properties>
              <l7:Property key="revision">
                <l7:LongValue>1</l7:LongValue>
              </l7:Property>
              <l7:Property key="soap">
                <l7:BooleanValue>true</l7:BooleanValue>
              </l7:Property>
            </l7:Properties>
          </l7:PolicyDetail>
          <l7:Resources>
            <l7:ResourceSet tag="policy">
              <l7:Resource type="policy">&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;
&lt;wsp:Policy xmlns:L7p=&quot;http://www.layer7tech.com/ws/policy&quot; xmlns:wsp=&quot;http://
schemas.xmlsoap.org/ws/2002/12/policy&quot;&gt;
  &lt;wsp:All wsp:Usage=&quot;Required&quot;&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;============================&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;======= Published thru API Portal =======&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;======= Don&apos;t modify block starts =======&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;============================&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:ApiPortalIntegration&gt;
      &lt;L7p:ApiGroup stringValue=&quot;&quot;/&gt;
```

```
      &lt;L7p:ApiId stringValue=&quot;d77f1e16-c899-4556-8a16-09c283cb8f8d&quot;/&gt;
      &lt;L7p:PortalManagedApiFlag stringValue=&quot;L7p:ApiPortalManagedServiceAssertion&quot;/&gt;
    &lt;/L7p:ApiPortalIntegration&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;aHR0cDovL2xvY2FsaG9zdDo4MDgwL2VjaG8=&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;apiLocation&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;c29hcGFwaWRlbW8=&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;serviceUrl&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;JHtwb3J0YWwubWFuYWdlZC5zZXJ2aWNlLmFwaUlkfQ==&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;counterName&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
        &lt;L7p:Encapsulated&gt;
      &lt;L7p:EncapsulatedAssertionConfigGuid stringValue=&quot;72093738-871a-45bd-b114-
ad3a61893ac0&quot;/&gt;
      &lt;L7p:Parameters mapValue=&quot;included&quot;&gt;
            &lt;/L7p:Parameters&gt;
    &lt;/L7p:Encapsulated&gt;
        &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;============================&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;======= Don&apos;t modify block ends =======&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;============================&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
          &lt;L7p:HttpRoutingAssertion&gt;
        &lt;L7p:FailOnErrorStatus booleanValue=&quot;false&quot;/&gt;
        &lt;L7p:ProtectedServiceUrl stringValue=&quot;
${apiLocation}${param.uri}${request.url.query}&quot;/&gt; &lt;!-- &lt;soap:address location=&quot;http://
www.webservicex.net/stockquote.asmx&quot; /&gt; --&gt;
        &lt;L7p:ProxyPassword stringValueNull=&quot;null&quot;/&gt;
        &lt;L7p:ProxyUsername stringValueNull=&quot;null&quot;/&gt;
        &lt;L7p:RequestHeaderRules httpPassthroughRuleSet=&quot;included&quot;&gt;
          &lt;L7p:Rules httpPassthroughRules=&quot;included&quot;&gt;
            &lt;L7p:item httpPassthroughRule=&quot;included&quot;&gt;
              &lt;L7p:Name stringValue=&quot;Cookie&quot;/&gt;
            &lt;/L7p:item&gt;
            &lt;L7p:item httpPassthroughRule=&quot;included&quot;&gt;
              &lt;L7p:Name stringValue=&quot;SOAPAction&quot;/&gt;
            &lt;/L7p:item&gt;
          &lt;/L7p:Rules&gt;
        &lt;/L7p:RequestHeaderRules&gt;
        &lt;L7p:RequestParamRules httpPassthroughRuleSet=&quot;included&quot;&gt;
          &lt;L7p:ForwardAll booleanValue=&quot;true&quot;/&gt;
          &lt;L7p:Rules httpPassthroughRules=&quot;included&quot;/&gt;
        &lt;/L7p:RequestParamRules&gt;
        &lt;L7p:ResponseHeaderRules httpPassthroughRuleSet=&quot;included&quot;&gt;
          &lt;L7p:Rules httpPassthroughRules=&quot;included&quot;&gt;
```

```
            &lt;L7p:item httpPassthroughRule=&quot;included&quot;&gt;
              &lt;L7p:Name stringValue=&quot;Set-Cookie&quot;/&gt;
            &lt;/L7p:item&gt;
          &lt;/L7p:Rules&gt;
        &lt;/L7p:ResponseHeaderRules&gt;
        &lt;L7p:SamlAssertionVersion intValue=&quot;2&quot;/&gt;
      &lt;/L7p:HttpRoutingAssertion&gt;
      &lt;/wsp:All&gt;
&lt;/wsp:Policy&gt;</l7:Resource>
            </l7:ResourceSet>
          </l7:Resources>
        </l7:Policy>
      </l7:Resource>
    </l7:Item>
    <l7:Item xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
      <l7:Name>soapapidemo-edit</l7:Name>
      <l7:Id>b2286ad6a8243794af20976095151300</l7:Id>
      <l7:Type>SERVICE</l7:Type>
      <l7:Resource>
        <l7:Service id="b2286ad6a8243794af20976095151300" xmlns:l7="http://ns.l7tech.com/2010/04/gateway-
management">
          <l7:ServiceDetail folderId="ddb84c6f397d7dbd3cca71d3043f019c"
 id="b2286ad6a8243794af20976095151300">
            <l7:Name>soapapidemo-edit</l7:Name>
            <l7:Enabled>true</l7:Enabled>
            <l7:ServiceMappings>
              <l7:HttpMapping>
                <l7:UrlPattern>/soapapidemo*</l7:UrlPattern>
                <l7:Verbs>
                                      <l7:Verb>POST</l7:Verb>
                                  </l7:Verbs>
              </l7:HttpMapping>
            </l7:ServiceMappings>
            <l7:Properties>
              <l7:Property key="internal">
                <l7:BooleanValue>false</l7:BooleanValue>
              </l7:Property>
              <l7:Property key="soap">
                <l7:BooleanValue>true</l7:BooleanValue>
              </l7:Property>
              <l7:Property key="tracingEnabled">
                <l7:BooleanValue>false</l7:BooleanValue>
              </l7:Property>
              <l7:Property key="wssProcessingEnabled">
                <l7:BooleanValue>false</l7:BooleanValue>
              </l7:Property>
              <l7:Property key="property.portalID">
                <l7:StringValue>d77f1e16-c899-4556-8a16-09c283cb8f8d</l7:StringValue>
              </l7:Property>
              <l7:Property key="property.internal.portalAPIEnabled">
                <l7:StringValue>true</l7:StringValue>
              </l7:Property>
              <l7:Property key="property.portalModifyTS">
```

```
                <l7:StringValue>1557167227011</l7:StringValue>
              </l7:Property>
                          </l7:Properties>
          </l7:ServiceDetail>
          <l7:Resources>
            <l7:ResourceSet tag="policy">
              <l7:Resource type="policy">&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;
&lt;wsp:Policy xmlns:L7p=&quot;http://www.layer7tech.com/ws/policy&quot; xmlns:wsp=&quot;http://
schemas.xmlsoap.org/ws/2002/12/policy&quot;&gt;
  &lt;wsp:All wsp:Usage=&quot;Required&quot;&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;------ Portal Created Fragment . Do not Modify ------&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;------- Encass has a route in it. ------&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:SetVariable&gt;
      &lt;L7p:Base64Expression stringValue=&quot;ZmFsc2U=&quot;/&gt;
      &lt;L7p:VariableToSet stringValue=&quot;override_template_routing&quot;/&gt;
    &lt;/L7p:SetVariable&gt;
    &lt;L7p:Include&gt;
      &lt;L7p:PolicyGuid stringValue=&quot;812ed196-c315-4e92-b630-b5c64c5c043c&quot;/&gt;
      &lt;L7p:PolicyName stringValue=&quot;Portal Service Preface&quot;/&gt;
    &lt;/L7p:Include&gt;
    &lt;L7p:Include&gt;
      &lt;L7p:PolicyGuid stringValue=&quot;d77f1e16-c899-4556-8a16-09c283cb8f8d&quot;/&gt;
    &lt;/L7p:Include&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;------ Portal Created Fragment . Do not Modify ------&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;L7p:CommentAssertion&gt;
      &lt;L7p:Comment stringValue=&quot;------ This routing path will be executed if using override
 template routing ------&quot;/&gt;
    &lt;/L7p:CommentAssertion&gt;
    &lt;wsp:OneOrMore wsp:Usage=&quot;Required&quot;&gt;
      &lt;L7p:ComparisonAssertion&gt;
        &lt;L7p:CaseSensitive booleanValue=&quot;false&quot;/&gt;
        &lt;L7p:Expression1 stringValue=&quot;${override_template_routing}&quot;/&gt;
        &lt;L7p:Expression2 stringValue=&quot;false&quot;/&gt;
        &lt;L7p:Predicates predicates=&quot;included&quot;&gt;
          &lt;L7p:item binary=&quot;included&quot;&gt;
            &lt;L7p:CaseSensitive booleanValue=&quot;false&quot;/&gt;
            &lt;L7p:RightValue stringValue=&quot;false&quot;/&gt;
          &lt;/L7p:item&gt;
        &lt;/L7p:Predicates&gt;
      &lt;/L7p:ComparisonAssertion&gt;
      &lt;wsp:All wsp:Usage=&quot;Required&quot;&gt;
        &lt;L7p:SetVariable&gt;
          &lt;L7p:Base64Expression stringValue=&quot;JHtyZXF1ZXN0Lmh0dHAudXJpfQ==&quot;/&gt;
          &lt;L7p:VariableToSet stringValue=&quot;param.uri&quot;/&gt;
        &lt;/L7p:SetVariable&gt;
        &lt;L7p:Regex&gt;
          &lt;L7p:AutoTarget booleanValue=&quot;false&quot;/&gt;
```

```
          &lt;L7p:OtherTargetMessageVariable stringValue=&quot;param.uri&quot;/&gt;
          &lt;L7p:PatternContainsVariables booleanValue=&quot;true&quot;/&gt;
          &lt;L7p:Regex stringValue=&quot;/${serviceUrl}&quot;/&gt;
          &lt;L7p:RegexName stringValue=&quot;process uri&quot;/&gt;
          &lt;L7p:Replace booleanValue=&quot;true&quot;/&gt;
          &lt;L7p:Replacement stringValue=&quot;&quot;/&gt;
          &lt;L7p:Target target=&quot;OTHER&quot;/&gt;
        &lt;/L7p:Regex&gt;
        &lt;wsp:OneOrMore wsp:Usage=&quot;Required&quot;&gt;
          &lt;wsp:All wsp:Usage=&quot;Required&quot;&gt;
            &lt;L7p:HttpRoutingAssertion&gt;
              &lt;L7p:FailOnErrorStatus booleanValue=&quot;false&quot;/&gt;
              &lt;L7p:ProtectedServiceUrl stringValue=&quot;
${apiLocation}${param.uri}${request.url.query}&quot;/&gt;
              &lt;L7p:ProxyPassword stringValueNull=&quot;null&quot;/&gt;
              &lt;L7p:ProxyUsername stringValueNull=&quot;null&quot;/&gt;
              &lt;L7p:RequestHeaderRules httpPassthroughRuleSet=&quot;included&quot;&gt;
                &lt;L7p:Rules httpPassthroughRules=&quot;included&quot;&gt;
                  &lt;L7p:item httpPassthroughRule=&quot;included&quot;&gt;
                    &lt;L7p:Name stringValue=&quot;Cookie&quot;/&gt;
                  &lt;/L7p:item&gt;
                  &lt;L7p:item httpPassthroughRule=&quot;included&quot;&gt;
                    &lt;L7p:Name stringValue=&quot;SOAPAction&quot;/&gt;
                  &lt;/L7p:item&gt;
                &lt;/L7p:Rules&gt;
              &lt;/L7p:RequestHeaderRules&gt;
              &lt;L7p:RequestParamRules httpPassthroughRuleSet=&quot;included&quot;&gt;
                &lt;L7p:ForwardAll booleanValue=&quot;true&quot;/&gt;
                &lt;L7p:Rules httpPassthroughRules=&quot;included&quot;/&gt;
              &lt;/L7p:RequestParamRules&gt;
              &lt;L7p:ResponseHeaderRules httpPassthroughRuleSet=&quot;included&quot;&gt;
                &lt;L7p:Rules httpPassthroughRules=&quot;included&quot;&gt;
                  &lt;L7p:item httpPassthroughRule=&quot;included&quot;&gt;
                    &lt;L7p:Name stringValue=&quot;Set-Cookie&quot;/&gt;
                  &lt;/L7p:item&gt;
                &lt;/L7p:Rules&gt;
              &lt;/L7p:ResponseHeaderRules&gt;
              &lt;L7p:SamlAssertionVersion intValue=&quot;2&quot;/&gt;
            &lt;/L7p:HttpRoutingAssertion&gt;
            &lt;L7p:SetVariable&gt;
              &lt;L7p:Base64Expression stringValue=&quot;JHtyZXNwb25zZS5odHRwLnN0YXR1c30=&quot;/&gt;
              &lt;L7p:VariableToSet stringValue=&quot;portal.analytics.response.code&quot;/&gt;
            &lt;/L7p:SetVariable&gt;
          &lt;/wsp:All&gt;
          &lt;wsp:All wsp:Usage=&quot;Required&quot;&gt;
            &lt;L7p:SetVariable&gt;
              &lt;L7p:Base64Expression stringValue=&quot;JHtyZXNwb25zZS5odHRwLnN0YXR1c30=&quot;/&gt;
              &lt;L7p:VariableToSet stringValue=&quot;portal.analytics.response.code&quot;/&gt;
            &lt;/L7p:SetVariable&gt;
            &lt;L7p:SetVariable&gt;
              &lt;L7p:Base64Expression stringValue=&quot;VW5hYmxlIHRvIHJvdXRlIHRvIEFQSS4=&quot;/&gt;
              &lt;L7p:VariableToSet stringValue=&quot;errorMsg&quot;/&gt;
            &lt;/L7p:SetVariable&gt;
```

```
            &lt;wsp:OneOrMore wsp:Usage=&quot;Required&quot;&gt;
              &lt;L7p:ComparisonAssertion&gt;
                &lt;L7p:CaseSensitive booleanValue=&quot;false&quot;/&gt;
                &lt;L7p:Expression1 stringValue=&quot;${portal.analytics.response.code}&quot;/&gt;
                &lt;L7p:ExpressionIsVariable booleanValue=&quot;false&quot;/&gt;
                &lt;L7p:Operator operatorNull=&quot;null&quot;/&gt;
                &lt;L7p:Predicates predicates=&quot;included&quot;&gt;
                  &lt;L7p:item dataType=&quot;included&quot;&gt;
                    &lt;L7p:Type variableDataType=&quot;string&quot;/&gt;
                  &lt;/L7p:item&gt;
                  &lt;L7p:item stringLength=&quot;included&quot;&gt;
                    &lt;L7p:Max intValue=&quot;-1&quot;/&gt;
                    &lt;L7p:Min intValue=&quot;1&quot;/&gt;
                  &lt;/L7p:item&gt;
                &lt;/L7p:Predicates&gt;
              &lt;/L7p:ComparisonAssertion&gt;
              &lt;L7p:SetVariable&gt;
                &lt;L7p:Base64Expression stringValue=&quot;NDA4&quot;/&gt;
                &lt;L7p:VariableToSet stringValue=&quot;portal.analytics.response.code&quot;/&gt;
              &lt;/L7p:SetVariable&gt;
            &lt;/wsp:OneOrMore&gt;
            &lt;L7p:FalseAssertion/&gt;
          &lt;/wsp:All&gt;
        &lt;/wsp:OneOrMore&gt;
        &lt;L7p:SetVariable&gt;
          &lt;L7p:Base64Expression stringValue=&quot;JHtyZXF1ZXN0LnJvdXRpbmdUb3RhbFRpbWV9&quot;/&gt;
          &lt;L7p:VariableToSet stringValue=&quot;portal.analytics.routingTotalTime&quot;/&gt;
        &lt;/L7p:SetVariable&gt;
        &lt;L7p:ExportVariables&gt;
          &lt;L7p:ExportedVars stringArrayValue=&quot;included&quot;&gt;
            &lt;L7p:item stringValue=&quot;portal.analytics.response.code&quot;/&gt;
            &lt;L7p:item stringValue=&quot;portal.analytics.routingTotalTime&quot;/&gt;
          &lt;/L7p:ExportedVars&gt;
        &lt;/L7p:ExportVariables&gt;
      &lt;/wsp:All&gt;
    &lt;/wsp:OneOrMore&gt;
  &lt;/wsp:All&gt;
&lt;/wsp:Policy&gt;</l7:Resource>
            </l7:ResourceSet>
                          <l7:ResourceSet tag="wsdl">
                <l7:Resource type="wsdl">&lt;wsdl:definitions xmlns:wsdl=&quot;http://schemas.xmlsoap.org/
wsdl/&quot; xmlns:ns1=&quot;http://org.apache.axis2/xsd&quot; xmlns:ns=&quot;http://c.b.a&quot;
 xmlns:wsaw=&quot;http://www.w3.org/2006/05/addressing/wsdl&quot; xmlns:http=&quot;http://
schemas.xmlsoap.org/wsdl/http/&quot; xmlns:xs=&quot;http://www.w3.org/2001/XMLSchema&quot;
 xmlns:soap=&quot;http://schemas.xmlsoap.org/wsdl/soap/&quot; xmlns:mime=&quot;http://
schemas.xmlsoap.org/wsdl/mime/&quot; xmlns:soap12=&quot;http://schemas.xmlsoap.org/wsdl/soap12/&quot;
 targetNamespace=&quot;http://c.b.a&quot;&gt;
    &lt;wsdl:documentation&gt;Calculator&lt;/wsdl:documentation&gt;
    &lt;wsdl:types&gt;
        &lt;xs:schema attributeFormDefault=&quot;qualified&quot; elementFormDefault=&quot;qualified&quot;
 targetNamespace=&quot;http://c.b.a&quot;&gt;
            &lt;xs:element name=&quot;add&quot;&gt;
                &lt;xs:complexType&gt;
```

```
                    &lt;xs:sequence&gt;
                        &lt;xs:element minOccurs=&quot;0&quot; name=&quot;n1&quot;
 type=&quot;xs:int&quot; /&gt;
                        &lt;xs:element minOccurs=&quot;0&quot; name=&quot;n2&quot;
 type=&quot;xs:int&quot; /&gt;
                    &lt;/xs:sequence&gt;
                &lt;/xs:complexType&gt;
            &lt;/xs:element&gt;
            &lt;xs:element name=&quot;addResponse&quot;&gt;
                &lt;xs:complexType&gt;
                    &lt;xs:sequence&gt;
                        &lt;xs:element minOccurs=&quot;0&quot; name=&quot;return&quot;
 type=&quot;xs:int&quot; /&gt;
                    &lt;/xs:sequence&gt;
                &lt;/xs:complexType&gt;
            &lt;/xs:element&gt;
        &lt;/xs:schema&gt;
    &lt;/wsdl:types&gt;
    &lt;wsdl:message name=&quot;addRequest&quot;&gt;
        &lt;wsdl:part name=&quot;parameters&quot; element=&quot;ns:add&quot; /&gt;
    &lt;/wsdl:message&gt;
    &lt;wsdl:message name=&quot;addResponse&quot;&gt;
        &lt;wsdl:part name=&quot;parameters&quot; element=&quot;ns:addResponse&quot; /&gt;
    &lt;/wsdl:message&gt;
    &lt;wsdl:portType name=&quot;CalculatorPortType&quot;&gt;
        &lt;wsdl:operation name=&quot;add&quot;&gt;
            &lt;wsdl:input message=&quot;ns:addRequest&quot; wsaw:Action=&quot;urn:add&quot; /&gt;
            &lt;wsdl:output message=&quot;ns:addResponse&quot; wsaw:Action=&quot;urn:addResponse&quot; /
&gt;
        &lt;/wsdl:operation&gt;
    &lt;/wsdl:portType&gt;
    &lt;wsdl:binding name=&quot;CalculatorSoap11Binding&quot; type=&quot;ns:CalculatorPortType&quot;&gt;
        &lt;soap:binding transport=&quot;http://schemas.xmlsoap.org/soap/http&quot;
 style=&quot;document&quot; /&gt;
        &lt;wsdl:operation name=&quot;add&quot;&gt;
            &lt;soap:operation soapAction=&quot;urn:add&quot; style=&quot;document&quot; /&gt;
            &lt;wsdl:input&gt;
                &lt;soap:body use=&quot;literal&quot; /&gt;
            &lt;/wsdl:input&gt;
            &lt;wsdl:output&gt;
                &lt;soap:body use=&quot;literal&quot; /&gt;
            &lt;/wsdl:output&gt;
        &lt;/wsdl:operation&gt;
    &lt;/wsdl:binding&gt;
    &lt;wsdl:binding name=&quot;CalculatorSoap12Binding&quot; type=&quot;ns:CalculatorPortType&quot;&gt;
        &lt;soap12:binding transport=&quot;http://schemas.xmlsoap.org/soap/http&quot;
 style=&quot;document&quot; /&gt;
        &lt;wsdl:operation name=&quot;add&quot;&gt;
            &lt;soap12:operation soapAction=&quot;urn:add&quot; style=&quot;document&quot; /&gt;
            &lt;wsdl:input&gt;
                &lt;soap12:body use=&quot;literal&quot; /&gt;
            &lt;/wsdl:input&gt;
            &lt;wsdl:output&gt;
```

```
                &lt;soap12:body use=&quot;literal&quot; /&gt;
            &lt;/wsdl:output&gt;
        &lt;/wsdl:operation&gt;
    &lt;/wsdl:binding&gt;
    &lt;wsdl:binding name=&quot;CalculatorHttpBinding&quot; type=&quot;ns:CalculatorPortType&quot;&gt;
        &lt;http:binding verb=&quot;POST&quot; /&gt;
        &lt;wsdl:operation name=&quot;add&quot;&gt;
            &lt;http:operation location=&quot;add&quot; /&gt;
            &lt;wsdl:input&gt;
                &lt;mime:content type=&quot;text/xml&quot; part=&quot;parameters&quot; /&gt;
            &lt;/wsdl:input&gt;
            &lt;wsdl:output&gt;
                &lt;mime:content type=&quot;text/xml&quot; part=&quot;parameters&quot; /&gt;
            &lt;/wsdl:output&gt;
        &lt;/wsdl:operation&gt;
    &lt;/wsdl:binding&gt;
    &lt;wsdl:service name=&quot;Calculator&quot;&gt;
        &lt;wsdl:port name=&quot;CalculatorHttpsSoap11Endpoint&quot;
 binding=&quot;ns:CalculatorSoap11Binding&quot;&gt;
            &lt;soap:address location=&quot;https://156.56.179.164:9443/services/
Calculator.CalculatorHttpsSoap11Endpoint/&quot; /&gt;
        &lt;/wsdl:port&gt;
        &lt;wsdl:port name=&quot;CalculatorHttpSoap11Endpoint&quot;
 binding=&quot;ns:CalculatorSoap11Binding&quot;&gt;
            &lt;soap:address location=&quot;http://156.56.179.164:9763/services/
Calculator.CalculatorHttpSoap11Endpoint/&quot; /&gt;
        &lt;/wsdl:port&gt;
        &lt;wsdl:port name=&quot;CalculatorHttpSoap12Endpoint&quot;
 binding=&quot;ns:CalculatorSoap12Binding&quot;&gt;
            &lt;soap12:address location=&quot;http://156.56.179.164:9763/services/
Calculator.CalculatorHttpSoap12Endpoint/&quot; /&gt;
        &lt;/wsdl:port&gt;
        &lt;wsdl:port name=&quot;CalculatorHttpsSoap12Endpoint&quot;
 binding=&quot;ns:CalculatorSoap12Binding&quot;&gt;
            &lt;soap12:address location=&quot;https://156.56.179.164:9443/services/
Calculator.CalculatorHttpsSoap12Endpoint/&quot; /&gt;
        &lt;/wsdl:port&gt;
        &lt;wsdl:port name=&quot;CalculatorHttpsEndpoint&quot;
 binding=&quot;ns:CalculatorHttpBinding&quot;&gt;
            &lt;http:address location=&quot;https://156.56.179.164:9443/services/
Calculator.CalculatorHttpsEndpoint/&quot; /&gt;
        &lt;/wsdl:port&gt;
        &lt;wsdl:port name=&quot;CalculatorHttpEndpoint&quot;
 binding=&quot;ns:CalculatorHttpBinding&quot;&gt;
            &lt;http:address location=&quot;http://156.56.179.164:9763/services/
Calculator.CalculatorHttpEndpoint/&quot; /&gt;
        &lt;/wsdl:port&gt;
    &lt;/wsdl:service&gt;
&lt;/wsdl:definitions&gt;

</l7:Resource>
              </l7:ResourceSet>
                      </l7:Resources>
```

```
        </l7:Service>
      </l7:Resource>
    </l7:Item>
  </l7:References>
  <l7:Mappings>
    <l7:Mapping action="NewOrUpdate" srcId="d77f1e16c89945568a1609c283cb8f8d" type="POLICY"
 xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management"/>
    <l7:Mapping action="NewOrUpdate" srcId="b2286ad6a8243794af20976095151300" type="SERVICE"
 xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management"/>
  </l7:Mappings>
</l7:Bundle>
```

3. Update the API on the proxy:

```
curl -X PUT -H 'Content-Type: application/xml' \
-u {username}:{password} -d @api-bundle.xml \
https://{proxyHost}/restman/1.0/bundle
```

Example request

```
curl -X PUT -H 'Content-Type: application/xml' \
-u admin:password -d @soap-api-bundle_updated.xml \
https://ohgateway1-94.lvn.broadcom.net:8443/restman/1.0/bundle
```

Sample response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<l7:Item xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
    <l7:Name>Bundle mappings</l7:Name>
    <l7:Type>BUNDLE MAPPINGS</l7:Type>
    <l7:TimeStamp>2019-05-06T11:40:56.180-07:00</l7:TimeStamp>
    <l7:Link rel="self" uri="https://ohgateway1-94.lvn.broadcom.net:8443/restman/1.0/bundle"/>
    <l7:Resource>
        <l7:Mappings>
            <l7:Mapping action="NewOrUpdate" actionTaken="UpdatedExisting"
 srcId="d77f1e16c89945568a1609c283cb8f8d" targetId="d77f1e16c89945568a1609c283cb8f8d" targetUri="https://
ohgateway1-94.lvn.broadcom.net:8443/restman/1.0/policies/d77f1e16c89945568a1609c283cb8f8d" type="POLICY"/>
            <l7:Mapping action="NewOrUpdate" actionTaken="UpdatedExisting"
 srcId="b2286ad6a8243794af20976095151300" targetId="b2286ad6a8243794af20976095151300" targetUri="https://
ohgateway1-94.lvn.broadcom.net:8443/restman/1.0/services/b2286ad6a8243794af20976095151300" type="SERVICE"/
>
        </l7:Mappings>
    </l7:Resource>
</l7:Item>
```

4. Update the API deployment status:

```
curl -X PUT -H 'Authorization: Bearer {token}' -H 'Content-Type:application/json;charset=utf-8' \
https://{portalApiHost}/{tenantId}/deployments/1.0/apis/{apiUuid}/proxies/{proxyUuid} \
-d '{"status": "DEPLOYED"}'
```

Undeploy a Scripted Deployment

Follow these steps:

1. Retrieve API delete bundle:

```
curl -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/api-management/1.0/apis/
{apiUuid}/bundle?type=delete
```

Example request

```
curl -k -H 'Authorization: Bearer 1313cea7-fc46-41d0-b983-b4a717121cb2' https://apim-ssg-
soapsupport.app.dev1.w2.saasqa.ca.com:443/tenantsoapsupport/api-management/1.0/apis/d77f1e16-
c899-4556-8a16-09c283cb8f8d/bundle?type=delete \
-o soap-api-bundle_delete.xml
```

Content of soap-api-bundle_delete.xml

```
<l7:Bundle xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
    <l7:Mappings>
            <l7:Mapping action="Delete" srcId="b2286ad6a8243794af20976095151300" type="SERVICE"
 xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management"></l7:Mapping>
            <l7:Mapping action="Delete" srcId="d77f1e16c89945568a1609c283cb8f8d" type="POLICY"
 xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management"></l7:Mapping>
        </l7:Mappings>
</l7:Bundle>
```

2. Delete API on the proxy:

```
curl -X PUT -H 'Content-Type: application/xml' \
-u {username}:{password} -d @api-bundle.xml \
https://{proxyHost}/restman/1.0/bundle
```

Example request

```
curl -X PUT -H 'Content-Type: application/xml' \
-u admin:password -d @soap-api-bundle_delete.xml \
https://ohgateway1-94.lvn.broadcom.net:8443/restman/1.0/bundle
```

Example response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<l7:Item xmlns:l7="http://ns.l7tech.com/2010/04/gateway-management">
    <l7:Name>Bundle mappings</l7:Name>
    <l7:Type>BUNDLE MAPPINGS</l7:Type>
    <l7:TimeStamp>2019-05-06T11:53:47.083-07:00</l7:TimeStamp>
    <l7:Link rel="self" uri="https://ohgateway1-94.lvn.broadcom.net:8443/restman/1.0/bundle"/>
    <l7:Resource>
        <l7:Mappings>
            <l7:Mapping action="Delete" actionTaken="Deleted" srcId="b2286ad6a8243794af20976095151300"
 targetId="b2286ad6a8243794af20976095151300" targetUri="https://ohgateway1-94.lvn.broadcom.net:8443/
restman/1.0/services/b2286ad6a8243794af20976095151300" type="SERVICE"/>
            <l7:Mapping action="Delete" actionTaken="Deleted" srcId="d77f1e16c89945568a1609c283cb8f8d"
 targetId="d77f1e16c89945568a1609c283cb8f8d" targetUri="https://ohgateway1-94.lvn.broadcom.net:8443/
restman/1.0/policies/d77f1e16c89945568a1609c283cb8f8d" type="POLICY"/>
        </l7:Mappings>
    </l7:Resource>
</l7:Item>
```

3. Delete deployment:

```
curl -X DELETE -H 'Authorization: Bearer {token}' https://{portalApiHost}/{tenantId}/deployments/1.0/apis/
{apiUuid}/proxies/{proxyUuid}
```

Example request

```
curl -X DELETE -H 'Authorization: Bearer 1313cea7-fc46-41d0-b983-b4a717121cb2' \
https://apim-ssg-soapsupport.app.dev1.w2.saasqa.ca.com:443/tenantsoapsupport/deployments/1.0/apis/
d77f1e16-c899-4556-8a16-09c283cb8f8d/proxies/845b69bc-0aa6-425c-91d3-99a37bca0221
```

Troubleshoot API Deployments

This article describes how to troubleshoot the following issues:

Portal Deployer is Not Receiving Deployment Events (On Demand API Deployments)

Symptoms

For On Demand API deployments, the deployment is stuck in a "Pending Deployment "state or "Pending Undeployment" state that does not resolve itself.

Solution

1. In the Policy Manager, restart the log by toggling the `portal.deployer.enabled` cluster property to `true` .
2. View the Gateway logs from within the Policy Manager or from the filesystem from the Gateway node log (*/opt/SecureSpan/Gateway/node/default/var/logs/ssg_X_0.log*).
   For more information, see "View Logs for the Gateway" section in the Gateway documentation.
3. To get the API out of its "Pending Deployment" state, redeploy the API by making the redeploy API call.
   For more information about how to make this API call, see REST API On-Demand Deployment.

Error When Deploying an API

Symptoms

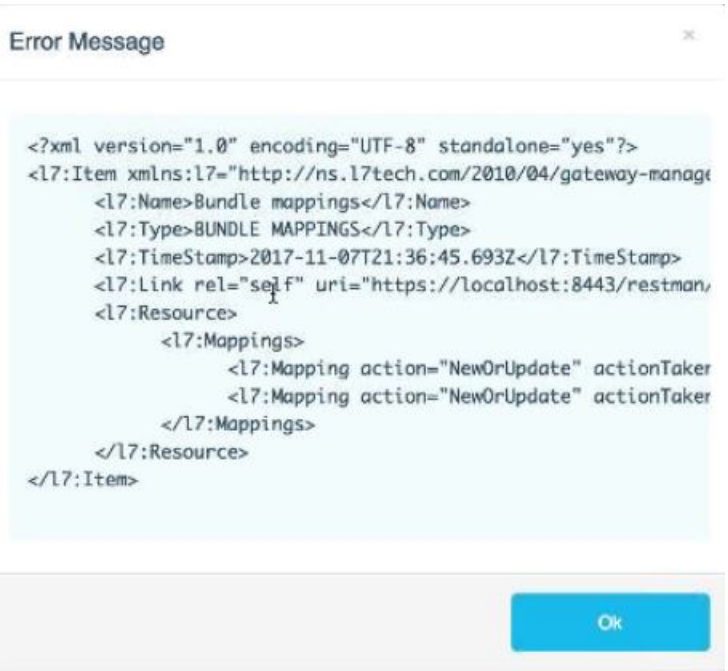
The API Details page displays an error for an API deployment.

Solution

View the details of an error by clicking the red icon of error status.

The following graphic displays an On Demand API deployment that contains an error status:

The following graphic displays the details of an example error message:



Failed Connecting to Broker Error (On Demand API Deployments)

Symptoms

For On Demand API deployments, the deployment is stuck in a "Pending Deployment" state. Gateway logs display the following error:

```
Could not connect to the message broker - check your DNS configurations
```

Solution

- Ensure that the DNS is configured correctly. Ensure that `broker.mycompany.com` is added to the `/etc/hosts` file.
- To get the API out of its Pending Deployment state, you need to redeploy the API by making the redeploy API call.

Service Cannot be Found Error

Symptoms

For On Demand API deployments, the API Details page displays an error for an API deployment. When the user selects the Error link, the message contains a reference to Service Not Found.

Solution

Ensure that RESTMAN is installed on the Gateway. See the "Rest Management API" section in the Gateway documentation.

Manage API Monitoring Tests

Portal Admins and API Owners with permissions to edit the API can manage API monitoring tests by creating them in Runscope, by viewing test results and the date of the latest test run, by viewing test details, and by unlinking your API from the tests.

In this article:

Create an API Monitoring Test

You can create API monitoring tests in Runscope from your API in API Portal. Creating a test from your API in API Portal creates a link to the test in Runscope. Your API can be linked to only one API monitoring test in Runscope at any given time.

Prerequisites:

- The Runscope integration is enabled.
  For more information about how to enable this integration, see Enable Integrations.
- You are familiar with Runscope and you know the parameters that are required to create an API monitoring test.
- An application in Runscope has been created. Runscope assigns each application an access token. The Layer7 Live API Creator integration engine requires this access token to authenticate calls to the Runscope API in Runscope and to create the API monitoring test in Runscope from your API in API Portal.
  For more information:
  – About how to create applications in Runscope, see the Runscope documentation.
  – About Layer7 Live API Creator, see the Live API Creator documentation.
- The bucket associated to the API monitoring test you are creating includes at least one environment.
- You are a Portal Admin or an API Owner with permissions to edit the API for which you want to create an API monitoring test.
- Your API is not already linked to an API monitoring test.

Follow these steps:

1. Log in to Layer7 Live API Creator as a Portal Admin or an API Owner with permissions to edit this API.
2. Select **Manage**, **APIs**.
   A list of APIs appears on the **APIs** page.
3. Select the API for which you want to create an API monitoring test.
   The **Overview** tab opens.
4. Select the **Integrations** tab.
5. For **Access Token**, enter the access token for the application, and then click **Test**. An example of an access token is cadcb571-f3f4-4364-9401-f6e5056d3ad2.
   The access token is verified and the connection to Runscope from Layer7 Live API Creator is validated.
6. Click **Save**.
   The access token in saved.
7. Specify the following fields, and then click **Save & Create Test**:
   Bucket
   Specifies the bucket to which this API monitoring test belongs. The buckets that display are based on the access token that you entered.
   Environment
   Specifies the environment variables that are used to run this API monitoring test.
   Schedule
   Specifies the frequency, or interval, at which you want this API monitoring test to run.
   Values: every minute, every 15 minutes, every 30 minutes, every hour, every 6 hours, every day
   Test Steps Scope
   By default, the integration engine creates the API monitoring test for the endpoints that are defined in the Swagger. This setting specifies the depth at which you want the integration engine to run this API monitoring test. The integration engine enables only the endpoints that are applicable based on this setting. The other test steps for the other endpoints are skipped.
   Values:

  - GET requests without parameters
  - All GET requests
  - All requests

The integration engine creates the API monitoring test using the following conventions:

```
API Title from the Swagger - Timestamp
```

The API monitoring test runs at the interval that you specify.

View Test Result and the Date for Latest API Monitoring Test Run

You can view the test result (Green/Red) and the date the test ran in Layer7 Live API Creator for the latest API monitoring test run. This information displays on the **Integrations** tab.

The test result states are:

- Pass (green). The latest API monitoring test run passed.
- Fail (red). The latest API monitoring test run failed.

Prerequisites:

- The API monitoring test for which you want to view the test result and test details has been created.
- You are a Portal Admin or an API Owner with permissions to view the API for which you want to view test result and the date for the latest test run.

View API Monitoring Test Details

You can view the details for the API monitoring test from Layer7 Live API Creator and view the API monitoring test scripts, a history of test results, and other details in Runscope.

Prerequisites:

- The API monitoring test for which you want to view test details has been created.
- You are a Portal Admin or an API Owner with permissions to view the API for which you want to view test details.

From the **Integrations** tab for the API, do one of the following:

- To view the details for the API monitoring test from Layer7 Live API Creator, click **View Details**. To return to the test result for the latest API monitoring test run in Runscope, click **View Latest Result**.
- To view the API monitoring test scripts, a history of test results, and other details in Runscope, click the **View result in Runscope** link.
  The API monitoring test opens in another browser tab.
  For more information about API monitoring tests, see the Runscope documentation.

Unlink the API Monitoring Test from Your API

> **NOTE**
> Unlinking the API monitoring test from your API does not delete the test in Runscope.
>
> For more information about how to delete API monitoring tests, see the Runscope documentation.

Prerequisites:

- The API monitoring test for which you want to unlink has been created.
- You are a Portal Admin or an API Owner with permissions to edit the API for which you want to unlink the test.

Follow these steps:

1. From the **Integrations** tab for the API, click **View Details**.

The details for the API monitoring test display.
2. Click **Unlink Test**.

The API is unlinked from the API monitoring test in Runscope.

Manage API Documents

API documents are markdown content that you can add to your API. Portal Admins, API Owners, and Org Publishers (for APIs that are assigned to their organization) have Update API permissions and can manage these documents. Managing API documents includes adding them, editing them, reordering them, and deleting them. These documents help the consumers of your API, or Developers, discover and learn about it.

API documents supplement API discovery and are in addition to the Swagger API documentation that is on the **Spec** tab. For example, you can add documents to your API that cover your performance metrics, functional specs, best practices, and use cases. Categorize and group them in a hierarchy on the Documentation tab.

You manage, or interact, with documents in the navigation tree in API Management SaaS.

> **NOTE**
> If you have localization requirements, manage your non-English documents using API Hub.
> For more information about API Hub, see API Hub.

> **TIP**
> The following procedures describe how to manage API documents by way of API Management SaaS. You can also manage API documents by making calls to the Portal API (PAPI) `document-service` resource.
> For more information about the PAPI, see Portal API (PAPI).

Add a Document

Follow these steps:

1. Log in to API Management SaaS as a Portal Admin, API Owner, or Org Publisher (for APIs that are assigned to their organization).
2. From the menu bar, select **Manage**, **APIs**.
   A list of APIs appears on the APIs page.
3. Click the API for which you want to add a document.
   The **Overview** tab opens.
4. Select the **Documentation** tab.
   If the API includes documents, a list of documents appears. Otherwise, the list is empty.
5. Complete one of the following:
   – To add a top-level document, click the Add Document icon (the + icon).
6. Add markdown content to your document in the pane to the left, the edit pane.
   The edit pane is a markdown editor. The edit pane and the pane to the right, the preview pane, are a side-by-side view of the document. You can:
   – Copy and paste markdown into your document.
   – Format the content that you add using the options on the toolbar or using markdown syntax.
     For more information about markdown syntax, see the Markdown Guide.

The document is added to the list of documents for the API.

> **NOTE**
> You can view the URI for this newly-added document in the web browser's address bar.

Edit a Document

Follow these steps:

1. From the list of documents, click the document that you want to edit from the list of documents to select it, and then click **Edit** (the pencil icon). Your document opens in edit mode, and options on the toolbar display.
2. As required, edit the **Title**.

   NOTE
   The URI cannot be edited once a document is published.
3. Edit the markdown content in the document or add content to the document. You can format the content using the options on the toolbar.
   A preview of your changes shows up in the pane to the right, the preview pane.
4. Save your changes by clicking **Publish**.

Navigate the Wiki Document Tree

You can navigate the document tree using the tab and arrow keys on your keyboard. Move the selector to the document tree using the tab key, then use the arrow keys to navigate through the document tree.

Reorder a Document

You can reorder documents within the document tree, such as moving a document to be a child document of another document or reordering the list of documents.

From the list of documents, click and hold the document that you want to reorder, and then drag and drop it to the new location in the tree.

Delete a Document

Deleting a parent document also deletes the child documents within the document tree (recursive delete).

Follow these steps:

1. From the list of documents, click the document that you want to delete, and then click **Delete** (trash bin icon).
   The **Confirm Delete** window opens.
2. Confirm the deletion by clicking **Yes**.

Manage API Lifecycles and States

Portal Admins and API Owners can change the state of an API based on the following API lifecycles:

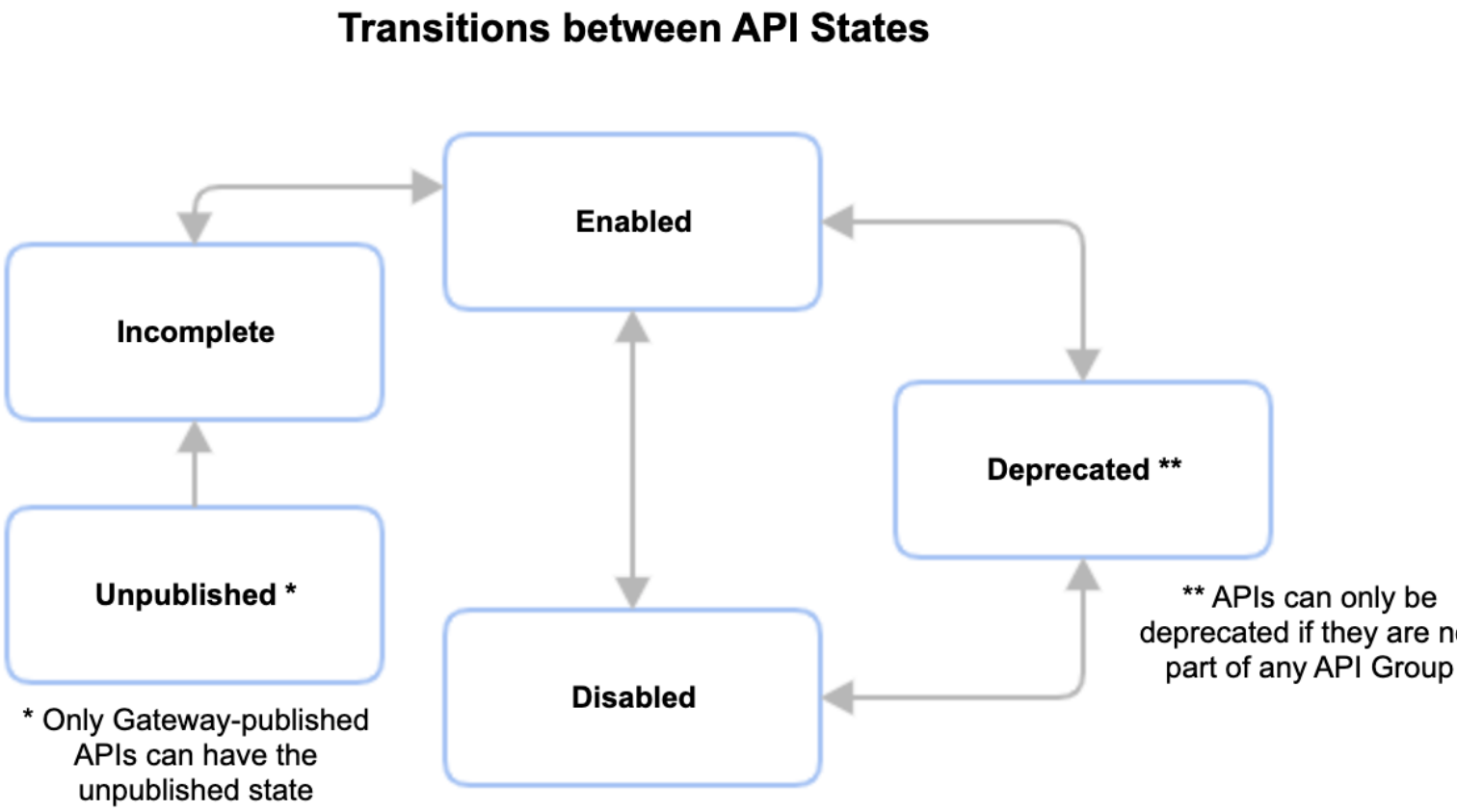
- Unpublished to Incomplete
- Incomplete to Enabled
- Deprecated to Enabled
- Disabled to Enabled
- Enabled to Deprecated
- Enabled to Disabled
- Deprecated to Disabled

Developers can add APIs and API groups that are in the Enabled state to their applications and applications can use them. When an API proxy administrator adds a *proxy-published* API to the API Portal, the state of the API is new. Developers can add only *proxy-published* APIs that are in the Enabled state to their applications.

After an API or API group that is in the Enabled state is added to an application, the application continues to use the API and API group in the following ways:

- Deprecated API groups continue to work with existing associated applications but applications cannot use them.
- You change the state of APIs that API groups use to Deprecated. For more information, see Manage API Groups.

> **NOTE**
> Even if a private API is enabled, Developers can add only those APIs to which they have access to their applications.



Change the State of an API

Follow these steps:

1. Log in as a Portal Admin or API Owner.
2. From the menu bar, select **Manage**, **APIs**.
   The **APIs** page appears.
3. Select the name of the API that you want to change the state.
   The **Overview** tab for the API appears.
4. Select **Actions**, **Edit API Details**.
   The **Details** page appears.
5. Change the state, and then select **Save & Next**.

The state of the API is changed.

Manage Applications

Applications are containers of related APIs in API Management SaaS. Org Admins and Developers can use applications to access those APIs while building their web/mobile application. Portal Admins and API Owners manage applications by adding, or registering, them to API Management SaaS, by editing them, by controlling which application use specific APIs, and by deleting them. Manage applications, for example, when testing your APIs and API Portal.

> **NOTE**
> You can also manage your applications by way of the Portal API (PAPI) or use this API in your scripts for managing applications.
>
> For more information about the PAPI, see Portal API (PAPI).

In this article:

Add an Application

All users, *except Developers*, can add, or register, applications to API Portal.

Follow these steps:

1. From the menu bar, select **Manage**, **Applications**.
   A list of applications appears on the **Applications** page.
2. Select **Add Application**.
   The **Details** page appears.
3. Provide details about the application. Select an existing organization from the **Selected Organization** drop-down list. Provide a unique application name and an optional description, and then select **Next**.
4. If the Portal Admin added custom fields for applications, then the **Custom Fields** page appears. Enter details for the custom field, and then click **Next**.
   The **API Management** page appears.
5. Add or remove available APIs and API groups to or from your application, and then select **Next**.
   In addition to the listed APIs and groups, you can search using the search field.
   Do the following:
   – **To remove a selected API or API group from the application**, select

     (the x icon) for the API or API group that you want to remove. The list of selected APIs and API groups is under the **Selected APIs** and **API Groups** section.
   – **To add an available API or API group to your application**, select

     (the plus icon) to the left of the API or API group that you want to add, and then accept the terms and conditions of the end-user license agreement (EULA). The list of available APIs and API groups is under the **Available APIs** (or **Available API Groups**) section.
     Prerequisite: You must have explicit access to the API or the API must belong to your organization.

   For more information about the effects of API lifecycles and states on your ability to add and remove APIs and API groups to and from your application, see Manage API Lifecycles and States.
   The **Authentication** page appears.
6. If any of the APIs that you have added to the application use OAuth, complete the following fields, and then select **Create**:
   – **Callback/Redirect URL(s)**
     Defines the callback/redirect URLs for your application. Separate multiple URLs using a comma.

```
https://{yourportalurl}/oauth2-redirect.html
```

- **Scope**
  Defines the OAuth scope parameters that specify the privileges that this application requires from the protected APIs. Separate parameters using a space.
- **Type**
  Defines the grant type for the OAuth-protected APIs that the application consumes.
  Values:
  - **None**.
  - **Public:** Defines that the OAuth-protected APIs that this application consumes use the Implicit grant type.
  - **Confidential:** Defines that the OAuth-protected APIs that this application consumes use the Confidential grant type.

  Default: None

The **Generate New Secret** window opens.

7. To generate a secret in hashed format, select **Create & Get Key**. Otherwise, to explicitly generate a less secure secret in plaintext format, select the **I want to use a non-secure plaintext key** checkbox, and then select **Create & Get Key**. The **Key** page appears. The application is successfully created. API Portal generates an API key for the application. The API key and shared secret are displayed in plaintext.
8. Do any of the following tasks, and then select **Done**:
   - Copy the shared secret or the API key to the clipboard.
   - Generate a new secret.
     For more information, see Generate a New Client Secret.

The application is added.

Edit an Application

All users can make the following changes to existing applications:

- Enable or disable the application.
- Edit the name and description of the application.
- Add or remove APIs and API groups to and from the application.
- Change the OAuth callback URL, scope value, and type.
- Generate a new shared secret.

Follow these steps:

1. On the **Applications** page, from the **Actions** drop-down for the application that you want to edit, select **Edit**.
   The **Details** page appears.
2. Edit the application name, enable or disable the application, or edit the public description, then select **Next**.
   #unique_77/unique_77_Connect_42_enableapikeys
3. If the Portal Admin added custom fields for applications, then the **Custom Fields** page appears. Edit the details for the custom field, and then click **Next**.
   The **API Management** page appears.
4. Add or remove available APIs and API groups to or from your application, and then select **Next**. By adding an API group to your application, you add the APIs that are contained within the group to your application. These APIs are enabled and public. If the APIs that are contained within the group are enabled but private, then the APIs belong to your organization and have been added to the account plan that your organization uses.

   > **NOTE**
   > If you have turned on API plans, you manage them instead of API groups on this page.
   > For more information about API plans, see Working with API Plans.

   In addition to the listed APIs and groups, you can search using the search field.

   Do the following steps:

– **To remove a selected API or API group from the application**, select

(the x icon) for the API or API group that you want to remove. The list of selected APIs and API groups is under the **Selected APIs** and **API Groups** section.

– **To add an available API or API group to your application**, select

(the plus icon) to the left of the API or API group that you want to add, and then accept the terms and conditions of the end-user license agreement (EULA). The list of available APIs and API groups is under the **Available APIs** (or **Available API Groups**) section.

Prerequisite: You must have explicit access to the API or the API must belong to your organization.

For more information about the effects of API lifecycles and states on your ability to add and remove APIs and API groups to and from your application, see Manage API Lifecycles and States.

The **Authentication & Keys** page appears.

5. Edit the following fields, and then select **Save Key**:

   NOTE
   Only Portal Admin or API Owner can add or remove keys.

   – **Callback/Redirect URL(s)**
   Defines the callback/redirect URLs for your application. Separate multiple URLs using a comma.
   `https://{yourportalurl}/oauth2-redirect.html`

   – **Scope**
   Defines the OAuth scope parameters that specify the privileges that this application requires from the protected APIs. Separate parameters using a space.

   – **Type**
   Defines the grant type for the OAuth-protected APIs that the application consumes.
   Values:
   - **None**.
   - **Public:** Defines that the OAuth-protected APIs that this application consumes use the Implicit grant type.
   - **Confidential:** Defines that the OAuth-protected APIs that this application consumes use the Confidential grant type.

   Default: None

   – **Client ID & Secret**
   Perform any of the following tasks:
   - Copy the API key to the clipboard.
   - If the shared secret is in plaintext format, copy it to the clipboard.
   - Generate a new secret by selecting **Generate New Secret**.
     For more information, see Generate a New Client Secret.

   For more information about how to manage API keys, see Manage API Keys.

6. Select **Done**.

Generate a New Client Secret

If an API key's shared secret is compromised, generate a new one, and then provide it to the application developer. Depending on your API Portal settings, API Portal generates secrets in the **Plaintext Secret** or **Hashed Secret** formats. For more information about hashed secrets, see Enable Hashed Client Secret.

WARNING
If you generate a new client secret, the API Proxy no longer accepts queries that use the old secret. The Developer must update the shared section in their web/mobile application so that their application can access the APIs.

Follow these steps:

1. With the application open in edit mode, select **Authentication & Keys**.
2. From the list of keys, expand on a Key Name to show the key details.
3. Select **Generate New Secret**.
   API Portal generates a new shared secret for the API key.
4. Select **Save Key**.

Locate your Applications

You can find and examine your applications from the **Applications** page. This page shows a list of the applications, the organization associated to the application, and the status of each application (Enabled or Disabled).

View an Application's Details

You can view various details of an application on the **Configuration**, **APIs**, and **Deployments** tabs within an application open in read-only mode. To view these tabs, on the **Applications** page, select the name of the application for which you want to view details. The **Configuration** tab displays by default.

View Details for an API Key

The **Configuration** tab shows a list of the API keys that have been added to the application, the status of each API key (Enabled or Disabled), and which API key is the default API key. Select to expand the API key row for which you want to view details, such as the API key (client ID), the shared secret (client secret), the status of the API key (Enabled or Disabled), and OAuth information.

View the APIs that have been added to an Application

The **APIs** tab shows tiles for each API that has been added to the application. Each API tile shows the API state (Enabled or Disabled) and the API version.

View Deployment Details for an API Key

The **Deployments** tab, like the **Configuration** tab, also shows a list of the API keys that have been added to the application, the status of each API key (Enabled or Disabled), and which API key is the default API key. Select to expand an API key row for which you want to view deployment details, such as tiles for each API proxy. Each API proxy tile shows the name of the API proxy, the date that the API proxy was last updated, how API Portal deploys the API key to the proxy (Automatic or On demand), the deployment status for the API key, and options to manage the API key deployment to API proxies.

For more information about the information that is available on this tab, including how to manage API key deployments to API proxies, see Deploy to Proxies using Portal.

For more information about how to manage API keys, see Manage API Keys.

Disable an Application

TIP

- Disabling an application disables all of its API keys.
- Re-enabling an application will re-enable the default key, while all other keys remain disabled. Ensure that you re-enable other keys individually.
- When a disabled application is re-enabled by an Org Admin, all other keys need to be re-enabled by Portal admin or API Owner.

Follow these steps:

1. View the application.
2. Click **Change** near the application Status, or select **Edit Application**
3. Change the status to **Disabled** and click **Save**.

Delete an Application

All users, *except Developers*, can delete applications.

Prerequisite: You have verified that the application does not include API keys that are deployed to on demand proxies.

Follow these steps:

1. With the application that you want to delete open in read-only mode (the **Configuration**, **APIs**, or **Deployments** tab is displayed), from the **Actions** menu, select **Delete Application**.
2. When prompted, select **Ok**.

The application is deleted.

Next Steps

Org Admins and Developers who want their web/mobile application to use the APIs that have been added to the application in API Portal must add the unique API key that API Portal auto-generated for the application to their web/mobile application. In addition, if their application uses OAuth, they must also add the shared secret to their application.

For more information about how Org Admins and Developers can work with applications, including how to add the API key and shared secret to their applications, see Work with Applications.

Manage API Keys

You manage API keys, including the default API key, by adding, editing existing, disabling, and deleting them.

As a Portal Admin or API Owner, you manage API keys, including the default API key, by adding, editing existing, disabling, and deleting them. When a user adds an application to API Portal, API Portal auto-generates the default API key for the application, a shared secret, and the other settings that you defined for the application. An API key is a unique identifier for the application. As a Portal Admin and API owner, you can define more granular access within a single application by adding API key/secret pairs for each designated consumer, and then deploy those API keys to different proxies. The application must include a default key and you can change which API key is the default key.

For more information about how to deploy API keys to proxies, see Manage API Key Deployments to Proxies.

Use Cases

The following examples highlight the different use cases for an application with multiple API keys:

Scenario: API Keys for Groups Within a Consumer

You have a shipping application that consumes several APIs to which multiple shipping agents require access. You want to group access to the APIs for multiple territories with designated proxies. Add a shipping application and add separate API keys to that application for each shipping agent. Deploy each API key to the proxy based on the shipping agent's designated territory.

Scenario: API Keys for Multiple Environments

You have registered proxies for your dev, QA, and prod environments in API Portal. You want to create API keys for these different environments. Add an application and add separate API keys to that application for each environment.

Scenario: API Key Rotation

You have business policies that require that you rotate your API keys periodically as a security measure. As part of business continuity, you can create another API key that works in parallel with the initial key before the switching over and disabling the initial key.

Add an API Key

Follow these steps:

1. Log in to API Management SaaS as a Portal Admin or API Owner.
2. Open the application in read-only mode (the **Configuration**, **APIs**, or **Deployments** tab is displayed).
3. From the **Actions** menu, select **Edit Keys**.
   A list of API keys display on the **Authentication & Keys** tab.
4. Select **Add Key**.
   Fields display to define the new API key.
5. Complete the following fields, and then select **Save Key**:
   - **Key Name**
     Defines the unique name for the API key. Give your API key an identifiable name that relates to its use. For example, create API keys for different environments, such as `paymentApp_devkey` for your development payment application.
   - **Default Key**
     Select this checkbox to assign this API key as the default key for this application.

     > **NOTE**
     > Assigning this API key as the default (selecting this checkbox), unassigns the current default key for the application. Applications can have only one assigned default key.

     Default: Cleared
   - **Status**
     Defines the status of the API key.
     Values: Enabled or Disabled
     Default: Enabled
   - **OAuth**
     If any of the APIs that you have added to the application use OAuth, complete the following fields:
     - **Callback/Redirect URL(s)**
       Defines the callback/redirect URLs for your API key. Separate multiple URLs using a comma.
       ```
       https://{yourportalurl}/oauth2-redirect.html
       ```
       Optional: Yes
     - **Scope**
       Defines the OAuth scope parameters that specify the privileges that this API key requires from the protected APIs. Separate parameters using a space.

Optional: Yes

- **Type**
  Defines the grant type for the OAuth-protected APIs that the API key consumes.
  Values:
  - **None**.
  - **Public:** Defines that the OAuth-protected APIs that this application consumes use the Implicit grant type.
  - **Confidential:** Defines that the OAuth-protected APIs that this application consumes use the Confidential grant type.

  Default: None
- **Client ID & Secret**

  NOTE
  API Portal generates the API key (client ID) and shared secret (client secret) when you save this API key.
- **Secret Type**
  Defines the format in which API Portal generates the secret for this API key. **Values:** Hashed or Plaintext
  Default: Hashed

  NOTE
  The plaintext format is less secure.

  The **Key** page appears. The application is successfully created. API Portal generates an API key for the application. The API key and shared secret are displayed in plaintext.

The API key is added to the application.

Disable an API Key

NOTE
The default API key cannot be disabled. As a workaround, you can disable the application, or set another key as default in order to disable the original key.

Follow these steps:

1. With the application open in read-only mode (the **Configuration**, **APIs**, or **Deployments** tab is displayed), from the **Actions** menu, select **Edit Keys**.
   A list of API keys display on the **Authentication & Keys** tab.
2. Select the name of the API key that you want to disable.
   The details for the API key display.
3. For **Status**, select **Disabled**.
4. Select **Save Key**.

The API key is disabled.

Delete an API Key

Deleting an API key deletes it from the application.

WARNING
If an Org Admin or Developer's web/mobile application uses the APIs that have been added to this application in API Portal by way of this API key, its access to the application in API Portal is lost. Consider disabling the API key instead.

Prerequisites:

- You have verified that the API key is not deployed to on demand proxies. For more information about how to undeploy an API key from an on demand proxy, see Deploy to Proxies using Portal or Deploy to Proxies using PAPI.
- The API key that you want to delete is not the default API key for the application.

Follow these steps:

1. On the **Applications** page, select the name of the application for which you want to delete an API key.
   The **Configuration** tab opens.
2. Select the **Authentication & Keys** tab.
   A list of API keys display.
3. Select the name of the API key that you want to delete.
   The details for the API key display.
4. Select **Delete Key**.
5. When prompted, select **Delete Key**.

The API key is deleted.

Manage API Key Deployments to Proxies

As a Portal Admin or API Owner, you can deploy an application's API keys to specific API proxies. These proxies represent specific environments and define the backend Gateways. Gateways process incoming requests from physical applications using proxies. You can define more granular access within a single application by adding API key/secret pairs for each designated consumer, and then deploy those API keys to different proxies.

For more information about API keys and how to manage them, Manage API Keys.

You can manage API key deployments to proxies using the following methods:

- Using API Portal.
- By making calls to the API Key Deployments resource for the Portal API (PAPI).
  For more information about the PAPI, see Portal API (PAPI).

Control How you Want your API Keys Deployed to Proxies

As a Portal Admin, you can control how API Portal deploys your API keys to proxies. You do this by setting the **Key Deployment Type** for the proxy.

You can set the key deployment type for a proxy to one of the following:

- **Automatic:** This is the default key deployment type for a proxy. Choose this type when you want API Portal to automatically deploy, undeploy, and redeploy application changes to all proxies. Changes include adding, editing, and deleting applications.
  This type is recommended in the following cases:
  – You want rapid iteration of development.
  – You want convenient and low-maintenance deployments.
  – You have development environments.
  Prerequisite: (To deploy API keys with hashed secrets to automatic proxies) You have OAuth Toolkit (OTK) version 4.4 or higher installed. For more information, see Enable Hashed Client Secret.
- **On Demand:** Choose this type when you want the Portal Admin to deploy, undeploy, and redeploy API keys to on demand proxies.
  This type is recommended in the following cases:
  – You have geographically-distributed environments.
  – You have user accepted testing (UAT) environments.
  – You want to stagger API key deployments to proxies across your development, testing, and production environments.
  Prerequisite: When a Portal Admin deploys, undeploys, and redeploys an API key to a proxy, the Portal Deployer client handles API key deployment, undeployment, and redeployment. Ensure that the API proxy administrator has installed and enabled the Portal Deployer modular assertion on the proxy during the enrollment or upgrade process.

For more information about how to troubleshoot issues with deploying API keys to on demand proxies, see Troubleshoot API Key Deployments.

For more information about the **Key Deployment Type** setting, see Manage Proxies.

Deploy to Proxies using Portal

This article provides information about how to manage API key deployments to proxies using API Portal.

In this article:

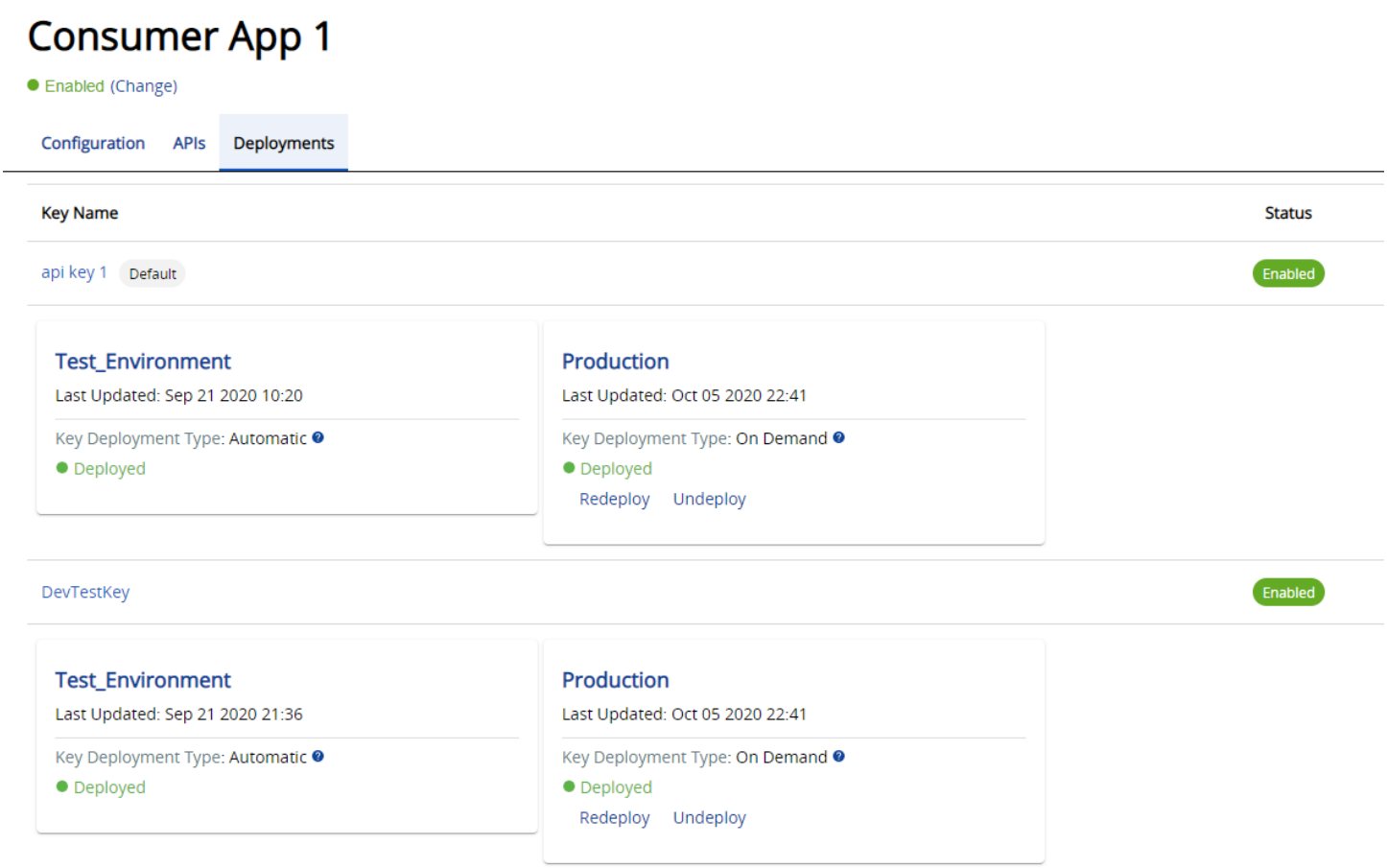
View API Key Deployment Details

You can view the following API key deployment details on the **Deployments** tab within an application:

> **NOTE**
> This tab displays when you have the application open in view-only mode.

- A list of the API keys that have been added to the application.
- For each API key:
  - The name of the API key.
  - The status of the API key:
    - Enabled
    - Disabled
  - Cards for each active proxy.
  - The deployment status for each API key:
    - Already deployed (the status is “Deployed”).
    - Pending deployment (the status is “Pending deployment”).
    - Not yet deployed (the status is “Not deployed”).
    - Pending undeployment (the status is “Pending undeployment”).
    - Error in the deployment (the status is “Deployment error” or "Validation error").
  - Messages about a deployment when there are errors in the deployment.

The following image shows an example of this tab:



Follow these steps:

1. Log in to API Portal as a Portal Admin.
2. From the menu bar, select **Manage**, **Applications**.
   A list of applications appears.
3. Select the name of the application for which you want to view application deployment details.
   The **Configuration** tab displays be default.
4. Select the **Deployments** tab.
   The active proxies display as cards on the page.

Manage API Key Deployments to On Demand Proxies

You can manage API key deployments by deploying, undeploying, and redeploy API keys to proxies.

Prerequisite: The key deployment type for the proxy is set to **On Demand**. For more information about this setting, see Manage Proxies.

Follow these steps:

1. On the **Deployments** tab, in a proxy's card, select from the following:
   - If the API key is not yet deployed to the proxy, click **Deploy**.
   - If the API key is already deployed to the proxy, is pending deployment, is pending undeployment, or has an error, click **Redeploy** to redeploy the API key to the proxy or **Undeploy** to undeploy the API key from the proxy.
2. Click **Ok** when prompted.

Deploy to Proxies using PAPI

This article provides information about how to manage application deployments to on demand proxies by making calls to the `API Key Deployments` resource for the Portal API (PAPI). You manage application deployments by deploying, undeploying, and redeploying applications.

For more information about the PAPI, see Portal API (PAPI).

In this article:

Verify the Prerequisites

Before you can manage application deployments by making calls to PAPI, ensure that you have met the following prerequisites:

- You have retrieved a valid OAuth 2.0 access token.
- You have installed the REST Management API (restman) on your proxy.
  For more information, see the Rest Management API section in the CA API Gateway documentation.

Retrieve the Application Deployment Type Setting for a Proxy

You can retrieve the application deployment type setting (on_demand or automatic) a proxy.

Issue a GET call to the following Proxies resource in the PAPI:

```
https://<papi_host>:<papi_port>/<tenantid>/deployments/0.1/proxies/{proxyUuid}/deployment-type
```

Example request:

```
curl -H 'Authorization: Bearer cde69bcc-3bed-44e0-af5b-c33fcb9020d5' \ https://apim-ssg-apim-uswest-
prod.app.prod.w2.dev.ca.com:443/atenant/deployments/0.1/proxies/39ae7b23-abba-4c6c-b057-b2ce97a76eb9/
deployment-type
```

Example response:

```
{
    "api": "AUTOMATIC",
    "application": "ON_DEMAND"
}
```

Set or Update the Application Deployment Type for a Proxy

Issue a PUT call to the following Proxies resource in the PAPI, specifying *applications* as the entity and including a payload:

```
https://<papi_host>:<papi_port>/<tenantid>/deployments/0.1/proxies/{proxyUuid}/deployment-type/{entity}
```

Payload to set or update to automatic:

```
{
    "api": "AUTOMATIC"
}
```

Payload to set or update to on-demand:

```
{
    "application": "ON_DEMAND"
}
```

Example request:

```
curl -H 'Authorization: Bearer cde69bcc-3bed-44e0-af5b-c33fcb9020d5' \ https://apim-ssg-apim-uswest-
prod.app.prod.w2.dev.ca.com:443/atenant/deployments/0.1/proxies/39ae7b23-abba-4c6c-b057-b2ce97a76eb9/
deployment-type/applications
```

Retrieve a List of the Proxies to Which an Application is Deployed

You can retrieve the list of the proxies to which the application is deployed and the application's deployment details, including the status, message, where the API key is deployed, and the application deployment type.

Issue a GET call to the following API Key Deployments resource in the PAPI:

```
https://<papi_host>:<papi_port>/<tenantid>/deployments/1.0/api-keys/{apiKey}/proxies
```

Example request:

```
curl -H 'Authorization: Bearer cde69bcc-3bed-44e0-af5b-c33fcb9020d5' \
  https://apim-ssg-apim-uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/api-keys/
l73b86486b0b324d2ba32c7a54b55bc796/proxies
```

Example response:

```
[
  {
      "applicationUuid": "ecbeb32f-710c-4dba-aa75-5e1fc52a9f8d",
      "apiKey": "l73b86486b0b324d2ba32c7a54b55bc796",
      "proxyUuid": "39ae7b23-abba-4c6c-b057-b2ce97a76eb9",
      "proxyName": "stageProxy",
      "lastTimeDeployed": 1585246140165,
      "status": "PENDING_DEPLOYMENT"
  }
  {
      "applicationUuid": "ecbeb32f-710c-4dba-aa75-5e1fc52a9f8d",
      "apiKey": "l73b86486b0b324d2ba32c7a54b55bc796",
      "proxyUuid": "0efddbdd-c3dc-43ce-8a64-6c0e1b0e1c68",
      "proxyName": "devProxy",
      "lastTimeDeployed": 1585250760192,
      "status": "PENDING_UNDEPLOYMENT"
  }
  {
      "applicationUuid": "ecbeb32f-710c-4dba-aa75-5e1fc52a9f8d",
      "apiKey": "l73b86486b0b324d2ba32c7a54b55bc796",
      "proxyUuid": "ce306f56-6cff-40cd-8d05-2c03a0c0a656",
      "proxyName": "ran_proxy_otk4.4",
      "lastTimeDeployed": 1585257888085,
      "status": "DEPLOYED"
  }
]
```

Retrieve a Specific Application Deployment to a Proxy

You can retrieve a specific application deployment to a proxy, including the status and message.

Issue a GET call to the following API Key Deployments resource in the PAPI:

```
https://<papi_host>:<papi_port>/<tenantid>/deployments/1.0/api-keys/{apiKey}/proxies/{proxyUuid}
```

Example request:

```
curl -H 'Authorization: Bearer cde69bcc-3bed-44e0-af5b-c33fcb9020d5' \
  https://apim-ssg-apim-uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/api-keys/
l73b86486b0b324d2ba32c7a54b55bc796/proxies/39ae7b23-abba-4c6c-b057-b2ce97a76eb9
```

Example response:

```
[
  {
       "applicationUuid": "ecbeb32f-710c-4dba-aa75-5e1fc52a9f8d",
       "apiKey": "l73b86486b0b324d2ba32c7a54b55bc796",
       "proxyUuid": "39ae7b23-abba-4c6c-b057-b2ce97a76eb9",
       "proxyName": "t1-proxy1",
       "lastTimeDeployed": 1585246140165,
      "status": "DEPLOYED"
  }
]
```

Deploy an Application to Proxies

Issue a POST call to the following API Key Deployments resource in the PAPI, which orchestrates and triggers the deployment process to the specific proxy:

```
https://<papi_host>:<papi_port>/<tenantid>/deployments/1.0/api-keys/{apiKey}/proxies
```

Example payload:

```
{
   "proxyUuid": "39ae7b23-abba-4c6c-b057-b2ce97a76eb9"
}
```

Example request:

```
curl -H 'Authorization: Bearer cde69bcc-3bed-44e0-af5b-c33fcb9020d5' \ https://apim-
ssg-apim-uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/api-keys/
l7023d322ee34740d0878bb75914c308a7/proxies -d '{ "proxyUuid": "39ae7b23-abba-4c6c-b057-
b2ce97a76eb9" }'
```

Redeploy an Application to Proxies

Issue a PUT call to the following API Key Deployments resource in the PAPI:

```
https://<papi_host>:<papi_port>/<tenantid>/deployments/1.0/api-keys/{apiKey}/proxies
```

Example payload:

```
{
   "status": "PENDING_DEPLOYMENT"
}
```

Example request:

```
curl -H 'Authorization: Bearer cde69bcc-3bed-44e0-af5b-c33fcb9020d5' \ https://apim-
ssg-apim-uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/api-keys/
l7023d322ee34740d0878bb75914c308a7/proxies -d '{ "status": "PENDING_DEPLOYMENT" }'
```

Redeploy an Application to a Specific Proxy

Issue a PUT call to the following API Key Deployments resource in the PAPI:

```
https://<papi_host>:<papi_port>/<tenantid>/deployments/1.0/api-keys/{apiKey}/proxies/{proxyUuid}
```

Example payload:

```
{
    "message": "string"
    "status": "PENDING_DEPLOYMENT"
}
```

Example request:

```
curl -H 'Authorization: Bearer cde69bcc-3bed-44e0-af5b-c33fcb9020d5' \ https://apim-ssg-apim-uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/api-keys/l7023d322ee34740d0878bb75914c308a7/proxies/39ae7b23-abba-4c6c-b057-b2ce97a76eb9
```

Undeploy an Application from a Specific Proxy

Issue a DELETE call to the following API Key Deployments resource in the PAPI:

```
https://<papi_host>:<papi_port>/<tenantid>/deployments/1.0/api-keys/{apiKey}/proxies/{proxyUuid}
```

After the application is deleted from the proxy, the deployment itself is deleted.

> **TIP**
> You can undeploy *and* delete the application from the proxy by issuing a DELETE request to this endpoint.

Example request:

```
curl -H 'Authorization: Bearer cde69bcc-3bed-44e0-af5b-c33fcb9020d5' \ https://apim-ssg-apim-uswest-prod.app.prod.w2.dev.ca.com:443/atenant/deployments/1.0/api-keys/8f6bc46c-a131-4388-b654-1e2b599b0ee9/proxies/39ae7b23-abba-4c6c-b057-b2ce97a76eb9
```

Troubleshoot API Key Deployments

This article describes how to troubleshoot the following issues:

Portal Deployer is Not Receiving Deployment Events (Deployments to On Demand Proxies)

The Portal Deployer client handles deployment, undeployment, and redeployment of API keys to proxies. It runs in the Gateway. If the connectivity between Gateway and API Portal experiences issues, you can resolve them.

Symptoms

The API key deployment is stuck in a "Pending Deployment" or a "Pending Undeployment" status that does not resolve itself.

Solution

- Restart the log by toggling the `portal.deployer.enabled` cluster property to `true` in the **Policy Manager**.
- View the Gateway logs from within the Policy Manager or from the filesystem from the Gateway node log (*/opt/SecureSpan/Gateway/node/default/var/logs/ssg_X_0.log*).

NOTE
See "View Logs for the Gateway" in the Gateway documentation.

- To get the API key out of the "Pending Deployment" status, redeploy the API key to the proxy by reissuing the redeploy API call.

Error When Attempting to Deploy an API Key with Hashed Shared Secrets (Deployments to Automatic Proxies)

Symptoms

You get the following error when trying to deploy an API key with a hashed shared secret to the proxy:


```
{

      "error":{
      "code":"ValidationException",
      "message":{
         "lang":"en",
         "value":"The request could not be completed due to data input errors."
      },
      "detail":{
         "errorCode":"483",
         "devErrorMessage":"The request could not be completed due to data input errors.",
         "userErrorMessage":"The request could not be completed due to data input errors.",
         "userErrorKey":"error.validation.entity",
         "validationErrors":[
            {
               "field":"application",
               "error":"Api key was deployed to proxies that have OTK version which does not support hashing
 secret. Please either use plain text secret or upgrade OTK version to 4.4 or higher of those proxies.",
               "key":"error.hashing.not.supported.by.proxies.generate.hash.secret.not.allowed"
            }
         ]
      }
   }
}
```


Solution

OAuth Toolkit (OTK) version 4.4 or higher is required. Ensure that you have this version installed.

Failed Connecting to Broker Error (Deployments to On Demand Proxies)

Symptoms

For API key deployments to On Demand proxies, the deployment is stuck in a "Pending Deployment" status. Gateway logs display a *Could not connect to the message broker - check your DNS configurations* error.

Solution

1. Ensure that you have configured your DNS server correctly. While editing the **hosts** file to point to the host serving API Portal from the system from which you want to access it, ensure that you have added `broker.mycompany.com` to the `/etc/hosts` file.
2. Redeploy the API key to the On Demand proxy.
   For more information:
   – About how to redeploy API keys to proxies using API Portal, see Deploy to Proxies using Portal.
   – About how to redeploy API keys to proxies using the Portal API (PAPI), see Deploy to Proxies using PAPI.

Work with Applications

Org Admins and Developers can build their applications and access the APIs that the application consumes using applications, which are containers of related APIs in API Management SaaS.

For more information about the roles and permissions for working with applications, see User Types, Roles and Permissions.

In this article:

Locate your Organization's Applications

You can find and examine your organization's applications.

Prerequisite: The Portal Admin has created an application for the Developer.

Follow these steps:

1. Log in to API Portal as an Org Admin or Developers.
2. From the menu bar, select **Manage**, **Applications**.
   A list of applications appears on the **Applications** page.

View the APIs Assigned to an Application

1. While logged in to API Portal as an Org Admin or Developers, from the **Applications** page, select the application for which you want to view the assigned APIs.
   The application opens in view-only mode.
2. Select the **APIs** tab.
   A list of the APIs that have been to the application are displayed.

Add an Application

Follow these steps:

1. While logged in to API Portal as an Org Admin, from the **Applications** page, select **Add Application**.
   The **Details** page appears.
2. Provide details about the application. Select an existing organization from the **Selected Organization** drop-down list. Provide a unique application name and an optional description, and then select **Next**.
3. If the Portal Admin added custom fields for applications, then the **Custom Fields** page appears. Enter details for the custom fields, and then click **Next**.
   The **API Management** page appears.
4. Add or remove available APIs and API groups to or from that application, and then select **Next**.
   In addition to the listed APIs and groups, you can search using the search field.
   Do the following:
   – **To remove a selected API or API group from the application**, select

     (the x icon) for the API or API group that you want to remove. The list of selected APIs and API groups is under the **Selected APIs** and **API Groups** section.
   – **To add an available API or API group to your application**, select

     (the plus icon) to the left of the API or API group that you want to add, and then accept the terms and conditions of

the end-user license agreement (EULA). The list of available APIs and API groups is under the **Available APIs** (or **Available API Groups**) section.
When you add an API group to your application, you add the APIs that are contained within the group to your application. These APIs are enabled and public. If the APIs that are contained within the group are enabled but private, then the APIs belong to your organization and have been added to the account plan that your organization uses.
Prerequisite: You must have explicit access to the API or the API must belong to your organization.

For more information about the effects of API lifecycles and states on your ability to add and remove APIs and API groups to and from your application, see Manage API Lifecycles and States.
The **Authentication** page appears.

5. If any of the APIs that you have added to the application use OAuth, complete the following fields, and then select **Create**:
   – **Callback/Redirect URL(s)**
     Defines the callback/redirect URLs for your application. Separate multiple URLs using a comma.
     `https://{yourportalurl}/admin/oauthCallback`
   – **Scope**
     Defines the OAuth scope parameters that specify the privileges that this application requires from the protected APIs. Separate parameters using a space.
   – **Type**
     Defines the grant type for the OAuth-protected APIs that the application consumes.
     Values:
     - **None**.
     - **Public:** Defines that the OAuth-protected APIs that this application consumes use the Implicit grant type.
     - **Confidential:** Defines that the OAuth-protected APIs that this application consumes use the Confidential grant type.

     Default: None

   The **Generate New Secret window** opens.
6. To generate a secret in hashed format, select **Create & Get Key**. Otherwise, to explicitly generate a less secure secret in plaintext format, select the **I want to use a non-secure plaintext key** checkbox, and then select **Create & Get Key**.
   The **Key** page appears. The application is successfully created. API Portal generates an API key for the application. The API key and shared secret are displayed in plaintext.
7. Do any of the following tasks, and then select **Done**:
   – Copy the shared secret or the API key to the clipboard.
   – Generate (or request) a new secret.
     For more information, see Edit an Application.

The application is added.

Edit an Application

You can make the following changes to an existing application:

- Enable or disable the application.
- Edit the name and public description of the application.
- Add and remove APIs and API groups to and from the application.
- Change the OAuth callback URL, scope value, and type.
- Generate a new shared secret, or, if the Portal Admin requires that they review and approve your requests to edit the application (the Edit Application Request Workflow setting is enabled), request a new shared secret.

Follow these steps:

1. From the **Applications** page, on the **Actions** drop-down for the application that you want to edit, select **Edit**.

The **Details** page appears.

2. Edit the application name, enable or disable the application, or edit the public description, and then select **Next**.

   TIP

   - Disabling an application disables all of its API keys.
   - Re-enabling an application will re-enable the default key, while all other keys remain disabled. Ensure that you re-enable other keys individually.
   - When a disabled application is re-enabled by an Org Admin, all other keys need to be re-enabled by Portal admin or API Owner.

3. If the Portal Admin added custom fields for applications, then the **Custom Fields** page appears. Edit the details for the custom field, and then click **Next**.
   The **API Management** page appears.
4. Add or remove APIs and API groups to and from your application. Accept the terms and conditions of the end-user license agreement (EULA). Select **Next**.

   NOTE
   A list of only those APIs to which you have access to add to your application are presented.

   The **Authentication & Keys** page appears.
5. You can do the following, and then click **Done**:
   - Edit the OAuth callback URL, scope value, and type.
   - View the API key and the shared secret.
   - Copy the API key or shared secret by clicking **Copy**.
   - Generate (or request) a new shared secret, for example, if the shared secret is compromised. Depending on API Portal settings, **Plaintext Secret** and/or **Hashed Secret** formats might be available.
     For more information about hashed secrets, see Enable Hashed Client Secret.

     WARNING
     When you generate a new shared secret, the API proxy no longer accepts queries that use the old secret. The Developer must update the shared secret in their web or mobile application so that it can access and use the APIs that the application consumes.

The changes to the application are saved.

Enable your Web/Mobile Application to Access the APIs Added to an Application in Portal

As you build your web/mobile application, add the unique API key and shared secret from the application in API Portal so that your web/mobile application can access and use those APIs. In addition, if your web/mobile application uses OAuth, add the shared secret to your web/mobile application.

Follow these steps:

1. From the **Applications** page, select the application for which you want to view the API key and shared secret details. The **Configuration** tab displays, showing a list of the API keys for the application.
2. Expand the API key that you require connection details.
3. Copy the **Client ID/API Key** and **Shared Secret**.
4. Add this information to your web/mobile application.

Delete an Application

Org Admins can delete applications. From the **Applications** page, on the **Actions** drop-down for the application that you want to delete, select **Delete**. The application is deleted.

NOTE
You can also delete an application with the application open, from the **Actions** menu, by selecting **Delete Application**.

Manage API EULAs

Portal Admins and API Owners can manage end-user license agreements (EULAs) in API Portal. When an Org Admin adds an API to their organization, they must agree to the EULA. All APIs published in API Portal require a EULA. So, before Portal Admins can add an API, a Portal Admin or API Owner must add a EULA to the API Portal.

> **NOTE**
>
> You can also manage your API EULAs by way of making calls to the `ApiEulas` resource for the Portal API (PAPI) or by using this API in your scripts for managing API EULAs.
>
> For more information about the Portal API, see Portal API (PAPI).

We recommend you to use the new `api-management/1.0/api-eulas` endpoint to perform EULA-related operations. See the latest PAPI swagger file to learn more.

In this article:

Add a EULA

You can add EULAs to API Portal.

Follow these steps:

1. Log in to API Portal as a Portal Admin or API Owner.
2. From the menu bar, select **Manage**, **EULAs**.
   The **API EULAs** page opens.
3. Select **Add API EULA**.
   The **Add API EULA** page opens.
4. Enter a unique name and the API EULA content, and then elect **Create**.

Examine a EULA

You can find EULAs from the **API EULAs** page. To examine the content of an API EULA, select the EULA.

Edit a EULA

> **CAUTION**
> Update EULA content at your own risk. If you have published APIs associated with a revised EULA, API consumers who agreed to the original end-user license agreement terms will NOT be notified by the Portal of any changes you may have made to those terms.

You can edit the name and content of a EULA and assign EULAs to APIs. The name must be unique.

Prerequisites:

- The EULA is not already assigned to APIs associated with an organization.
- The EULA is not already associated to an API that has been added to an application.

Follow these steps:

1. On the **API EULAs** page, on the **Actions** drop-down for the EULA that you want to edit, select **Edit**.
2. Edit the API EULA.
3. Select **Save**.

Delete a EULA

Prerequisite: The EULA is not associated with a published API.

Follow these steps:

1. Log in to API Portal as a Portal Admin or API Owner.
2. From the menu bar, select **Manage**, and then **EULAs**.
   The **API EULAs** page opens.
3. Select a EULA from the list.
   The **EULA Details** page opens.
4. Click the **Actions** button and select **Delete EULA**.

Manage Proxies

As a Portal Admin, you can manage proxies. Proxies represent specific environments and define the backend Gateways. Gateways process incoming requests from physical APIs and applications using the proxies.

In this article:

About Enhanced Synchronization

Beginning with the release of version 5.0.2 of the API Portal, API and API key deployment synchronization between the Portal and On-Premise proxies has been enhanced for improved reliability and scalability. This enhancement has also allowed the API Portal to capture and present richer data in the Proxy Details page for analysis and troubleshooting. To ensure that your Portal and proxy integration is taking full advantage of this enhanced synchronization, the following items and actions should be considered:

- The latest Portal Integration bundle is installed.
- The Portal Compatibility tile displays an 'Up to Date' message, confirming that the latest Portal Integration bundle is installed.

Modifying an existing proxy's configuration from On-Demand to Automatic synchronization ensures that the proxy will adopt the enhanced synchronization method. New proxy enrolments with Automatic synchronization will always adopt the enhanced synchronization method.

Add an API Proxy

Portal Admins typically add On-Premise proxies while setting up API Portal and add SaaS proxies after installing API Portal, when enrolling a Layer7 API Gateway.

Follow these steps:

1. Log in to API Portal as a Portal Admin.
2. From the menu bar, select **Manage, Proxies**.
   A list of API proxies display on the API Proxy page.
3. Select **Add Proxy**.
   The Details page appears.
4. Complete the following fields, and then select **Save & Next**:
   – **Proxy Name:** Defines the unique name for your proxy.
   – **API Deployment Type:** Control how API Portal deploys newly Portal-published APIs to this proxy by selecting the API deployment type. API Portal automatically deploys the changes that you make to existing APIs to the proxies regardless of the deployment type.
     Options:

- **Automatic:** Choose this type when you want API Portal to automatically deploy the API to all proxies as soon as you publish the API.
- **On Demand:** Choose this type when you want the Portal Admin to manage API deployments to proxies as needed.
- **Scripted:** Choose this type when you want to use existing Continuous Integration/ Continuous Development (CI/CD) processes and script the deployment to automate API deployment. Include calls to the `ApiDeployments` resource for the Portal API (PAPI) in your deployment script.

Default: Automatic
For more information about how to deploy APIs to proxies, see Manage API Deployments.

– **Key Deployment Type:** Control how API Portal deploys newly-added API keys to this proxy by selecting the key deployment type. API Portal automatically deploys the changes that you make to existing API keys to all proxies regardless of the deployment type.

Options:

- **Automatic:** Choose this type when you want API Portal to automatically deploy API keys to all proxies as soon as you add the API key to the application.
- **On Demand:** Choose this type when you want the Portal Admin or Org User with deployment permissions to manage API key deployments to proxies as needed.

Default: Automatic
For more information about how to deploy API keys to proxies, see Manage API Key Deployments to Proxies.

NOTE
You can also manage the key deployment type for proxies by making calls to the `Proxies` resource for the PAPI.
For more information about the PAPI, see Portal API (PAPI).

5. On the **Organization Assignment** tab, under **Organizations**, select or clear the checkboxes for the organizations that you want to assign to or un-assign from the proxy, and then click **Save**.
   After you assign an organization to a proxy, a Publisher within the organization can deploy an API they own or manage to that proxy.

   TIP
   You can also assign organizations to a proxy using the `Proxies` resource for the PAPI. However, you can un-assign organizations from the proxy only using API Portal.

   For information, see Manage Organization Assignments to Proxies using PAPI.

Edit an API Proxy

You can edit an API proxy, for instance:

- To change the name of the proxy. For example, to give it another name or to correct a spelling mistake.
- To edit the API or key deployment type.

  CAUTION
  Changing a deployment type can cause synchronization issues for Gateway-published APIs. To learn more about each type, see Manage API Deployments.
- To assign or un-assign an organization to or from the proxy.

Follow these steps:

1. Select **Proxies** from the Manage menu.
2. On the API Proxy page, select **Edit** next to the API proxy that you want to edit.
   The **Details** tab opens.
3. Edit the proxy name, the API deployment type, and the key deployment type, and then select **Save & Next**.
   The **Organization Assignment** tab opens.
4. In the **Organizations** section, assign organizations to the proxy. Select or clear the checkboxes for the organizations that you want to assign to or un-assign from the proxy, and then select **Save**.

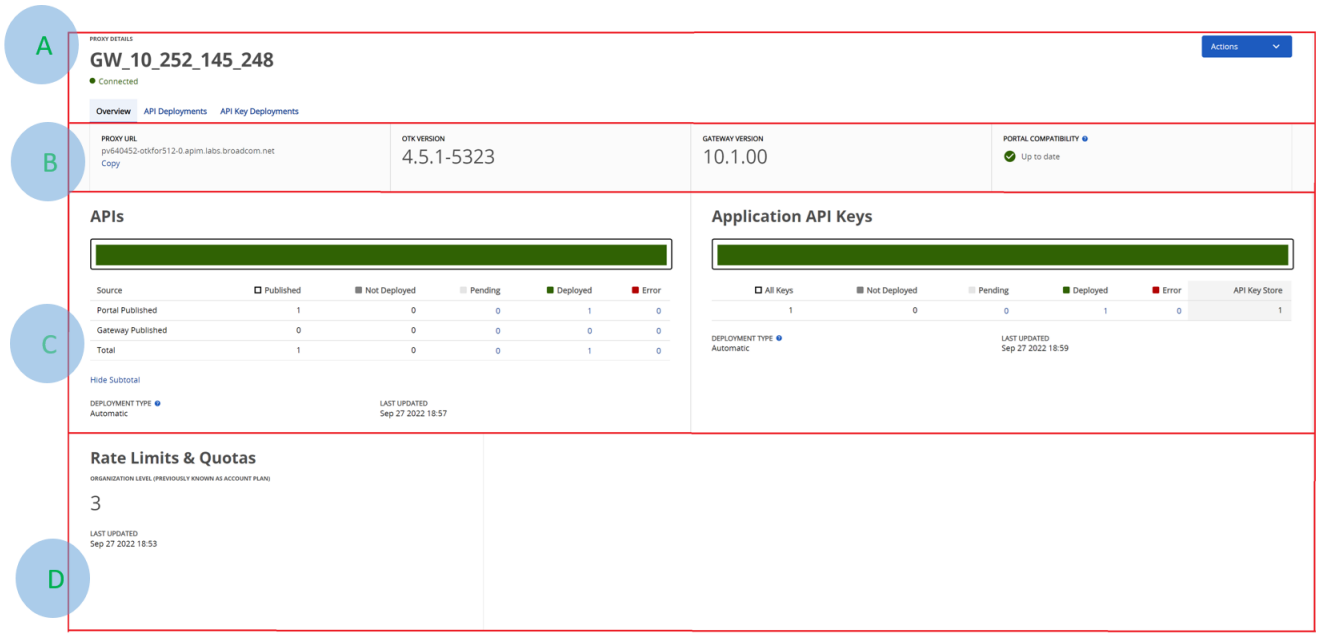
TIP
You can also assign organizations to the proxy using the PAPI. To un-assign an organization from the proxy, use API Portal.
For more information, see Manage Organization Assignments to Proxies using the PAPI.

View Proxy Details

The Proxy Details page shows Portal administrators how API Gateways are currently enrolled and configured to work with the API Portal and serves as a dashboard for Portal administrators who want to quickly understand the status of their proxies and effectively gain insights from all deployment activities managed by the API Portal. Proxy Details are broken down into three distinct tabs, each providing a unique view of your proxy deployment:

- Overview
- API Deployment
- API Key Deployments

Overview Tab



The following describes the Overview tab in four main areas A,B, C and D:

A - Header: Proxy Name, Connection Status, and Navigation Tabs

Proxy Name and Connection Status

In the heading area of the page, you can quickly identify the API proxy by its name at the top-left corner, followed by the connection status of the proxy which indicates whether it’s connected to the Portal (green) or disconnected from the Portal (red).

A connected status indicates that the Portal is able to send messages and synchronize with the proxy for API and key deployment. If a disconnected status is shown, contact your API Gateway administrator to investigate the connectivity issue.

TIP
The connection status does not appear unless the latest Portal Integration bundle is installed (see 'Portal Compatibility' under 'B - Basic Proxy Details'). Install the latest bundle to ensure you can view all the available data in the Proxy Details page.

Navigation Tabs

As required, you may navigate away from the Overview tab to view details for the API Deployments or API Key Deployments associated with the proxy by clicking their respective tabs.

Action Menu

Use the drop-down menu to edit or delete the proxy.

B - Proxy Basic Details

The first row of tiles display the following information about the proxy and its components:

- The API proxy URL
- The version of the OAuth Toolkit solution
- The version of the connected API Gateway
- Portal Compatibility: Lets you quickly determine if your proxy has the latest Portal Integration Bundle installed

Up to Date	The API Proxy is compatible with the currently installed version of the API Portal, provided the latest Portal Integration Bundle is installed on the proxy after a Portal upgrade.
Update Required	The installed Portal Integration Bundle is outdated and requires updating in order for you to optimize API and API Key deployments and continue using all of the Portal's available features. To learn how to update the Portal Integration Bundle, see Integrate On-Premise API Proxies.

C - Primary Deployment Information Tiles

The second row of information tiles display information about API management assets that are maintained from the Portal for the proxy, including APIs and API Key Deployments. Each of the two tiles show respective groupings (subtotals) of API and API Key counts by the following deployment statuses:

- Published (White)
- Pending (Light Gray)
- Not Deployed (Dark Gray)
- Deployed (Green)
- Error (Red)

Within the APIs tile, APIs are also grouped by their publishing source, and are either Portal-published or Gateway-published. Each tile also indicates the Deployment Type (Automatic or On-Demand) and the time and date of when the presented information was last updated on the screen.

NOTE

Why Are the API Key Counts Inconsistent Between the Overview and API Key Deployments Tabs?

The API Key counts in the Overview and API Key Deployments tabs may appear inconsistent or mismatched if a proxy was disconnected or an issue occurred during API key deletion.

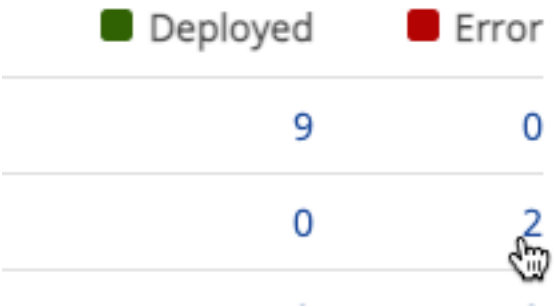
Stacked Horizontal Bar

As a visual aid, a stacked horizontal bar graph appears at the top of each tile, letting you quickly compare API or API key deployment statuses by color segmentation. In the example below, we can see that the majority of APIs are deployed (green) to the proxy.

Troubleshoot Deployments

When reviewing the Overview tab of the Proxy Details, seeing an API or API Key with an ‘Error’ status may signify a synchronization issue. Typically, this means that changes made to an API or Application have not been deployed to the proxy.

You may also click an API or API key count to drill into the finer details for those APIs or API keys in the API Deployments OR API Key Deployment tabs. In the example below, we’re curious to see which two Portal-published APIs have the ‘Error’ status and determine what the next steps are to deploy them. Clicking the URL leads us to the API Deployments tab that can offer this insight (see the API Deployments Tab section on this page).



D - Secondary Proxy Information Tiles: Rate Limits & Quotas

The secondary proxy information tiles provide the following information:

- The Rate Limits & Quotas tile displays information about the rate limits and quotas:
  - Portal displays the total number of rate limits and quotas in API Portal.

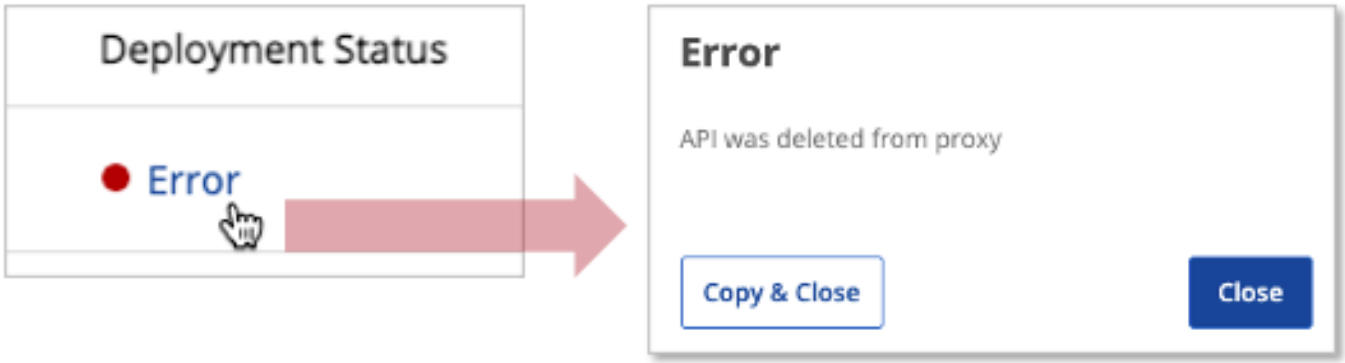
API Deployments Tab

The API Deployments tab lists all the APIs that are marked for deployment to the proxy. Each API is uniquely identified by its name, and given the following attributes organized in columns from left to right:

- State: Unpublished, Enabled, Disabled, Deprecated or Deleted - see Manage API Lifecycles and States Publish
- Source: Where the API was published from, either from the Portal or Gateway
- Last Deployed: The date and time the API was last deployed to the proxy Deployment
- Status: Deployed, Pending Deployment, Pending Undeployment, Error Undeploy, Error

To find out more about an API (e.g., its public description, documentation, location, etc.), you can drill into the individual APIs by clicking their linked names in the API Name column.

For any APIs with an Error status, click the error link to view the message describing the cause of the error.



Filtering and Sorting

You may apply a combination of any number of filters to view a select list of API deployments that interest you with the following drop-down filters at the top of the API list:

- State (Default is All States)
- Publish Sources (Default is All Publish Sources)
- Deployment Status (Default is Any Deployment Status)

You may sort your list by a last-deployed time-line of either ‘New to Old’ or ‘Old to New’.

API Key Deployments Tab

Structured similarly to the API Deployments tab, the API Key Deployments tab lists all the API keys that are available to applications associated with the proxy. Each API key is uniquely identified by its unique Key Name, and given the following attributes organized in columns from left to right:

- API Key: The unique identifier code used to authenticate a client attempting to connect to an API Key
- Status: Enabled or disabled
- Key Deployment Status: Deployed, Pending Deployment, Pending Undeployment, Error Undeploy, Error
- API Keystore: The number of API keys that currently reside in a keystore such as OTK.

To find out more about an API key (e.g., its Client Secret, OAuth attributes, associated API, etc.), you can drill into the individual API keys by clicking their linked names in the API Key Name column.

For any API keys with an Error status, click the error link to view the message describing the cause of the error.

Filtering and Sorting

You may apply a combination of any number of filters to view a select list of API keys that interest you with the following fields at the top of the API list:

- Key Name (Enter characters from Key names)
- API Key (Enter a key - Exact Match Only)
- Key Status (Select from Drop-down list - Default is Any Status)
- Deployment Status (Select from Drop-down list - Default is Any Deployment Status)

You may sort your list by a last-deployed time-line of either ‘New to Old’ or ‘Old to New’.

Delete an API Proxy

You can delete a proxy if you no longer need it, if it is causing problems, or if you added it by mistake.

When you delete an API proxy, all references to that proxy are removed from API Portal. Analytics data for that API proxy remain but is no longer accessible.

> **NOTE**
> You cannot delete the last enrolled API proxy.

Follow these steps:

1. Select **Proxies** from the Manage menu.
2. On the API Proxy page, select **Delete** next to the API proxy that you want to delete.
3. Select **Ok** to confirm the deletion.

Manage API Usage

> **NOTICE**
>
> **About Account Plans**
>
> To better serve Portal users and API publishers and to streamline the rate limiting capability across Organization, API per Organization, and APIs, Account Plans now fall under ‘Rate Limits and Quotas’ in the Manage menu of the API Portal as of Portal version 5.1.

API usage management falls under two categories in the API Portal:

- Rate Limits and Quotas (formerly 'Account Plans', account plans are now incorporated into Rate Limits and Quotas)
- API Plans

Rate Limits and Quotas

You can access and constrain the usage of APIs by defining **Rate Limits and Quotas** at two different assignment levels. As a Portal admin, can define assignments at the following levels:

- Organization
- API
- API per organization

When assigned to an organization, the rate limit and quota constrains the cumulative API usage for that organization. When the Rate Limits and Quotas are assigned at the API level, the access limits can be used as input to policies across all API usage during API publishing. When assigned to an API per organization, API usage constraints can be imposed to a specific API per a specific organization.

Portal Admins can manage the visibility of APIs at the organization level. For more information about API visibility and permissions, see Create and Set Permissions for APIs.

API Plans

API plans enable more granular control on how developers and applications within an organization can consume individual APIs. The API plan comprises rate limit and/or quota information, along with the public or private APIs to which these controls apply. You can also choose to which organizations an API plan applies, allowing you to set different access tiers for different organizations for the same APIs.

Used concurrently, you can manage manage consumption and tiered quota on your APIs using rate limits and quotas *and* API plans included within an application.

In this topic:

Rate Limits and Quotas vs API Plans

The following table highlights the similarities and differences between rate limits and quotas and API plans:

	Rate Limits and Quotas...	API Plans...
Organizations	Organization Assignment Level: Can be applied to different organizations. Restricts an organization to follow only one rate limit and quota. API Assignment Level: N/A API Per Organization Level: Restrict an organization to follow only one rate limit and quota for a specific API.	Must be linked to organizations and APIs to make the plan selectable when creating applications within those organizations. Enable an organization to use different API plans for different APIs and applications.
APIs	Organization Assignment Level: N/A API Assignment Level: Can be applied to different APIs. Each API can only have one rate limit and quota assigned to define the access limit. API Per Organization Assignment Level: Can be applied to different APIs. Each API can be given multiple Rate Limit and Quotas with the API per organization assignment level.	Enable an organization to use different API plans for different APIs and applications.
API Consumption	Organization Assignment Level: Constrain the cumulative usage limit for the organization under the rate limit and quota. Drives the overall access and quota of your organizations. API Assignment Level: Protect the backend and constrains the cumulative usage limit for the individual API. Drive the overall access and quota of the API across all consumers. API Per Organization Assignment Level: Protect the backend and constrain the cumulative usage limit for the individual API per organization. Drive the overall access and quota of the API per organization.	Constrain the consumption limit of individual APIs by developers and applications. Lets you manage access tiers for optimum consumption by organizations within your rate limit and quota.
Roles and Permissions	Portal Admins can add and edit rate limits and quotas for all assignment levels. API Owners can view rate limits and quotas at the organization assignment level. API publishers can select rate limits and quotas to constrain API access at the API or API per organization level,	Portal Admins can add and edit. Organization Admin and Developer can select API plans for individual applications, or propose selections to be approved by Admins or API Owners.

How Quotas and Rate Limits Work

Rate limits and quotas and API plans can use different quotas and rate limits.

Quota per day or month

Restricts the number of times that an application can query an API in a day or month.

- In rate limits and quotas, a quota limit of 10000 per day ensures that all activities of an organization do not exceed 10000 hits per day.
- In API plans, a quota limit of 10000 per day ensures that application hits for the API assigned to that API plan do not exceed 10000 hits per day.

Rate limit per second

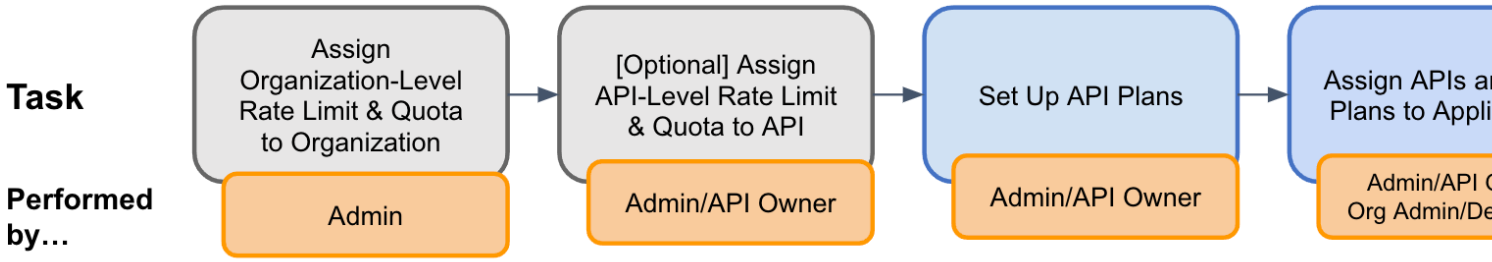
Restricts the number of times that an application can query an API in a second.

- In rate limits and quotas, a rate limit of 100 per second ensures that all activities do not query more than 100 times per second.
- In API plans, a rate limit of 100 per second ensures that application queries for the API assigned to that API plan do not exceed 100 times per second.

Following this logic, administrators can give each API a different quota and rate limit for different applications.

Getting Started with Rate Limits and Quotas and API Plans

The recommended workflow is as follows:



- If you do not have rate limits and quotas already configured from a previous deployment, see Manage Rate Limits and Quotas.
- If you are already using rate limits and quotas and want to incorporate API plans into your business logic, see Manage API Plans.

The following best practices are recommended if you have existing rate limits and quotas:

- Check and adjust the current rate limit and quota limit. Use the highest common denominator when assigning rate limit and quota at the organization level.
- Revisit your existing API-level rate limit(s) and quota(s) (if applicable) before defining your API Plan's rate limit and quota.
- If you have not already, enable request workflows. This enables the Admin or API Owner to accept or reject requests made by Org Admins and Developers to edit applications and select API plans. If you disable the request workflow, Org Admins and Developers can perform these tasks without approval.

Manage Rate Limits and Quotas

Rate limiting can protect your API or back-end resources from being overwhelmed with requests and improve general availability by limiting the rate of transactions. Similarly, quotas ensure API consumers (external customers or internal organization members) stay within the number of requests permitted within a predefined time period, thereby allowing your enterprise to effectively manage API usage.

In Portal, the quota specifies the maximum number of hits per day or month, while the rate limit specifies the maximum number of hits per second.

> **NOTICE**
>
> **About Account Plans**
>
> To better serve Portal users and API publishers and to streamline the rate limiting capability across Organization, API per Organization, and APIs, Account Plans now fall under ‘Rate Limits and Quotas’ in the Manage menu of the API Portal as of Portal version 5.1.

Portal Admins can add, edit, and delete rate limits and quotas at various assignment levels (i.e., API level or Organization level or API per Organization level). These rate limits and quotas can then be applied when creating new or editing existing APIs or Organizations.

Organization Level

Portal Admins can constrain an organization's API usage by assigning Rate Limits and Quotas to an organization. This effectively places a constraint on all API usage for an organization. The following implementation details apply:

- All organizations require a rate limit and quota assigned. Rate limits and quotas can be applied to multiple organizations, but each organization is limited to only one rate limit and quota.
- Organizations cannot exceed their assigned quota and rate limit regardless of how the hits are divided among an organization's applications and APIs.
- The configuration of a rate limit and quota can be applied to multiple organizations requiring the same access limit.

When a user self-registers for API Management SaaS, API Management SaaS assigns the Org Admin role and a new organization is created for the user. The `Bronze` default account plan (i.e., rate limit and quota) is automatically assigned to this organization.

API Level

Similar to rate limiting at the organization level, rate limits and quotas can be applied to APIs or groups of APIs. Regardless of which API consumer calls the API, the rate limit and cumulative quota is imposed on the API itself. Rate limiting at the API level can be used as input to policy templates to protect your API backend during API publishing.

The following implementation details apply:

- Rate limits and quotas are optional for an API. Rate limits and quotas can be applied to multiple APIs, but each API is limited to only one rate limit and quota.
- APIs cannot exceed their assigned quota and rate limit regardless of the source of hits or requests.

API Per Organization Level

Provides additional granularity for setting quota and rate limits that are applied to each API across all applications within a given organization.

The following implementation details apply:

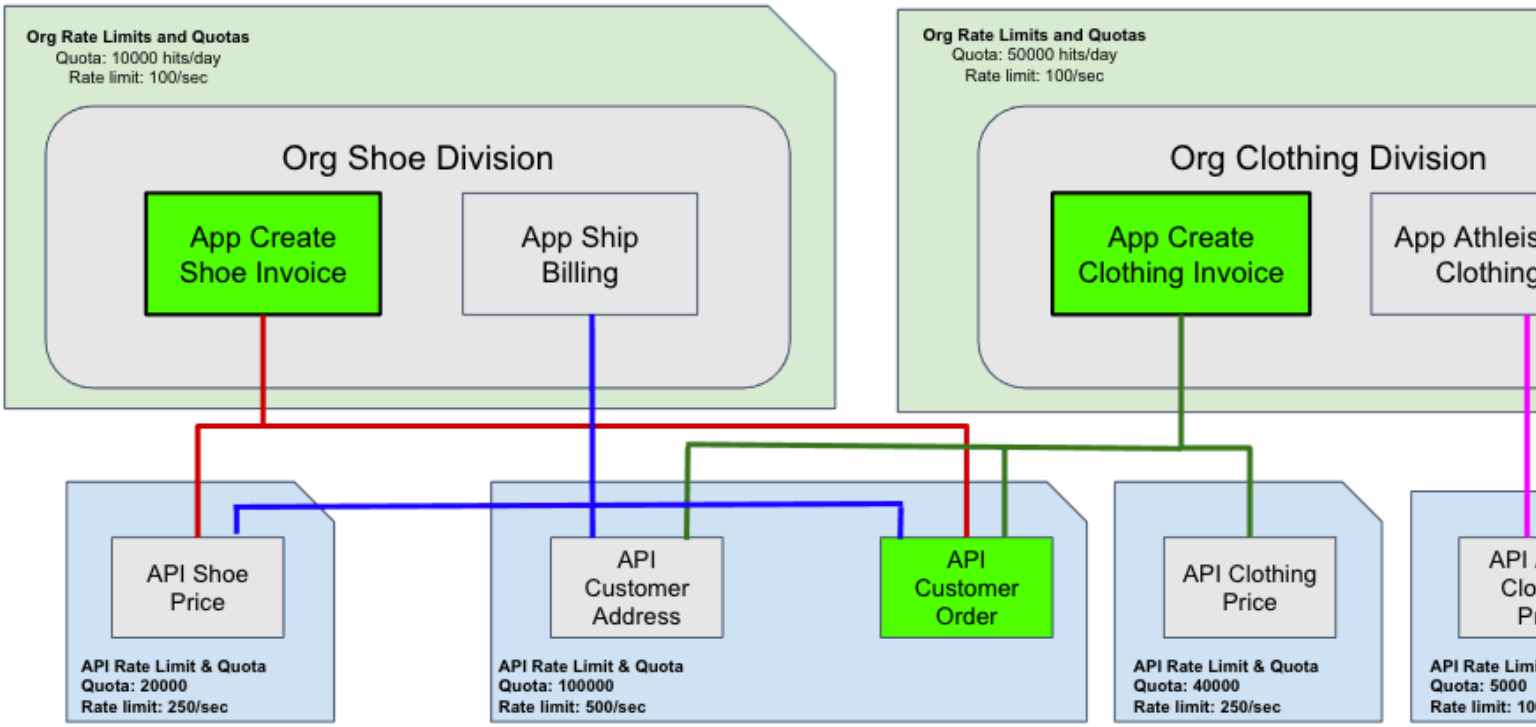
- The organization must be given visibility or access to the API that is being constrained.
- A rate limit and quota policy template (i.e., *l7.apim.system - Rate & Quota Policy Template - 2.0.1*) must be selected first when adding or editing an API
- You can assign a rate limit and quota with an 'API per organization' assignment level to each individual organization in the **Organization** tab of the API Details page.
- Each organization is limited to one 'API per organization' rate limit and quota.

Contents:

- How Rate Limit and Quota Works
- View a Rate Limit and Quota
- Add a Rate Limit and Quota
- Edit a Rate Limit and Quota
- Delete a Rate Limit and Quota
- About the Default API per Organization Plan

How Rate Limit and Quota Works

The following diagram illustrates the relationship between rate limits and quotas, organizations (including their applications), and APIs. For illustration purposes, the parent organization is a national apparel and shoe online retailer.



Two Organizations with Two Different Rate Limits and Quotas

The company contains both a shoe division and clothing division and are represented as separate organizations in the Portal. Each organization is assigned a different rate limit and quota (formerly known as Account Plan), Each unique rate limit and quota is used to govern the cumulative usage of each organization.

Multiple APIs with Different Rate Limits and Quotas

Customer Address and Customer Order APIs are assigned with identical rate limits and quotas to protect the retailer's API backend. Each API's rate limit and quota is an independent limit, used to govern the usage of each of these APIs across all organization and their applications. On the other hand, the Shoe Price, Clothing Price and Active Clothing Price APIs each have different rate limits and quotas assigned.

Rate Limiting in Action

With all the different rate limits and quotas applied to this retailer's rate limiting setup, how will the different rate limits and quotas assigned at the organization and API levels become impacted by a single API transaction? To illustrate, let's look at the Customer Order API.

When the Customer Order API is called once each by the Create Shoe Invoice application and Create Clothing Invoice application, the API hits will incur usage counts from:

- The quota defined at the organization level for the Shoe Division organization AND
- The quota defined at the organization level for the Clothing Division organization.
- The quota defined at the API level for the Customer Order API.

When there are multiple API hits calling the Customer Order API, when the lower rate limit defined either at the organization level or API level is met first, the API Gateway will start limiting the number of requests calling the API.

Throwing 'API Per Organization' Into the Mix

Taking a step further in granularity, if the retailer were to create additional sub-organizations for the Shoe division, such as 'Shoe Division West' and 'Shoe Division East', it can configure the Portal to discern the following usage constraints with the 'API per organization' for the Customer Order API:

- Shoe Division West shall be capped at a quota of 2500 hits a day
- Shoe Division East shall be capped at a quota of 4000 hits a day

Like the previous usage limit scenario involving the organization and API assignment levels only, when the lower rate limit defined at any of the three assignment levels is met first, the API Gateway will start limiting the number of requests calling the API. For example, even if Shoe Division West was assigned a quota of 10,000 hits per day at the organization level (which applies to all APIs it has access to), the organization would be prevented from hitting the Customer Order API for the 2501st time.

In effect, if you were to monetize an API, the 'API per organization' rate limit and quota assignment level allows for tier-pricing. For example, you may offer a free tier (i.e., a 'free' organization) and a premium tier (i.e., a 'premium' organization), with different rate limits and quotas for each.

View a Rate Limit and Quota

To view a rate limit and quota:

1. Log in to Layer7 API Developer Portal as a Portal Admin or API Owner.
2. From the menu bar, select **Manage** > **Rate Limits and Quotas**.
   The Rate Limits and Quotas page appears, showing various rate limit and quota configurations, and whether the rate limit and quota is assigned at the organization, API, or API per organization level.
3. Click the name of a rate limit and quota configuration to view the Rate Limit and Quota Details page for
   - A detailed view of its rate limit and quota settings and description in the **Overview** tab.
   - A list of APIs or organizations currently using the rate limit and quota configuration in the **APIs** or **Organizations** tab.
   - A list of APIs and any associated organizations with an assigned limit (applicable to the 'API per Organization' assignment level) in the **APIs and Organizations** tab.

To Apply a Filter:

From the top of the list on the Rate Limits and Quotas page, you can restrict the shown list of Rate Limits and Quotas by applying one or more of the following filters:

- Name (Keyword search)
- Level (All levels (default), API, or Organization, or API per Organization)
- Limit (All limits (default), Rate only, Quota only)

Add a Rate Limit and Quota

To add a rate limit and quota:

1. From the Rate Limits and Quotas page, select **Add Rate Limit and Quota**.
   The Create Rate Limit and Quota page opens.
2. Enter a unique Name for the Rate Limit and Quota. Select the API, Organization, or API per Organization option as the assignment level.
3. Select one or more of the following:
   - **Rate**. If selected, enter a value (between 1 and 2,000,000,000) in the Rate Limit box for the number of allowable endpoint hits per second for an API or organization application. For the API assignment level, you may also enter values for the following options:

    - **Spread Limit Window**. If given a value (seconds), this option allows a burst of requests to spread over a time window in seconds. If the limit or window is exceeded, the excess requests shall be throttled.
    - **Maximum Concurrency**. If given a value (number equal to 1 or higher), the number of concurrent requests cannot exceed the number specified.
  - **Quota**. If selected, enter a value (between 1 and 2,000,000,000) in the Quota box for the number of allowable endpoint hits per the given Quota interval for an API or organization application. In the Quota Interval box, select Hour, Day, or Month. For the organization assignment level, select Day or Month (hour is available only for the API assignment level).
4. (Optional) Enter a **Description** of the Rate Limit and Quota.
5. Click **Save**.

The Rate Limit and Quota configuration you've created can be readily applied to any new or existing API, or Organization, or API per Organization.

To learn how to apply a Rate Limit and Quota

- Using a policy template, see Edit and Delete APIs.
- To an organization, see Manage Organizations.

Edit a Rate Limit and Quota

> **IMPORTANT**
> Prior to changing the Rate Limit and Quota for an Organization or API or API per Organization, carefully review its impact on the affected organization or APIs.

To edit a rate limit and quota:

1. From the Rate Limits and Quotas page, select the Rate Limit and Quota configuration from the list by clicking its name. The Rate Limit and Quota Details page appears.
2. In the Actions drop-down list, select **Edit Rate Limit and Quota**.
   If the Rate Limit and Quota is attached to one or more APIs or organizations, a warning pop-up appears. To proceed further, click Edit Rate Limit and Quota in the pop-up.
3. Edit the name, limit settings, and description of an existing Rate Limit and Quota as required.

   IMPORTANT

   - Changes to API level rate limit and quotas that are currently used by APIs require a redeployment of those affected APIs to proxies that are configured for manual or scripted API deployment (see Deploy APIs to learn more).
   - Changes to organization level rate limit and quotas are automatically updated on all proxies.
   - While you may edit the existing values of the rate limit and quota, you may NOT change the assignment level (i.e., from API to organization or vice versa). If such an edit is required, create a new Rate Limit and Quota instead.

About the Default API per Organization Plan

This plan is applied as the default rate limit and quota to all APIs on a per organization basis. As required, a different rate limit and quota with an API per Organization assignment level may be assigned to an organization. While this default plan cannot be deleted or renamed, the rate limit and quota values may be edited. To disable this default plan, deselect both the rate limit and quota options within this plan.

Enforcement of this default plan will require the version 2.0.1 of the Rate & Quota Policy Template Gateway Bundle to be deployed.

Delete a Rate Limit and Quota

You can only delete Rate Limit and Quotas that are not assigned to an organization or API. You cannot delete the default Rate Limit and Quota (i.e., the Bronze plan).

To delete a rate limit and quota:

1. From the Rate Limits and Quotas page, select the Rate Limit and Quota configuration from the list by clicking its name.
   The Rate Limit and Quota Details page appears.
2. In the Actions drop-down list, select Delete Rate Limit and Quota.
   The Confirm Delete pop-up appears.
3. Click Delete to confirm the removal of the Rate Limit and Quota.

Manage API Plans

> **IMPORTANT**
> To start using the enhanced synchronization method for API plans, ensure you have installed the latest update of the Portal Integration Bundle on the API proxy. For the latest version of the bundle supported by the Portal, see Compatibility Matrix.

API plans control the rates at which your applications can access individual APIs, both private and public. Portal Admins and API Owners can add, edit, and delete API plans.

Portal Admins, API Owners, Org Admins, and Developers can:

- View the details for an API plan, such as the description and rate and quota limits.
- Choose an API plan for APIs to which their application subscribes.
  For more information, see View and Choose API Plans.

In this topic:

For information about how to create, edit, and delete API plans, see Working with API Plans.

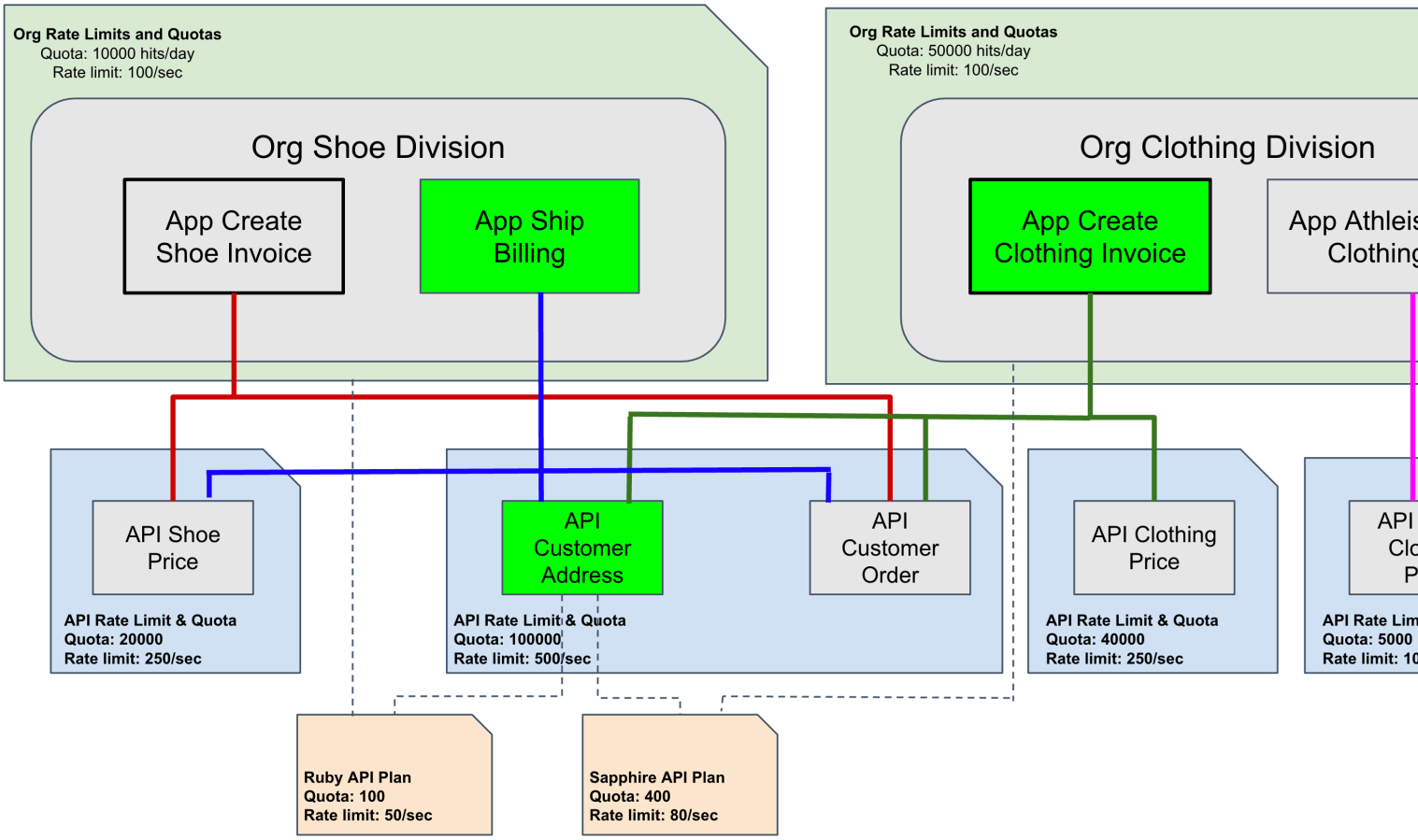
Prerequisites

Before you can manage API plans, ensure that you have met the following prerequisites:

- You are running the following product versions or higher:
  - Gateway 10.0
  - OTK 4.4
- There are no active API groups added to any applications.
- The Portal Admin has turned on API plans for the tenant.
  For more information, see Working with API Plans.

How API Plans Work with Rate Limits and Quotas

API plans can be defined for the Organization and API to provide more granular rate limit and quota configurations at the Application API key level. The following diagram, based on the Rate Limit and Quota diagram, illustrates how API plans may complement an existing Rate Limit and Quota configuration as part of an overall rate limiting strategy.



In this example, two different API plans are defined for the same API:

- The Ruby API plan is assigned to the Customer Address API for the Shoe Division organization
- The Sapphire API plan is assigned to the Customer Address API for the Clothing Division organization

At the application level, the following API plan-assignment scenarios are also possible:

- The Ship Billing application created for the Shoe Division organization using the Customer Address API can be assigned the Ruby API Plan.
- The Create Clothing Invoice application created for the Clothing Division organization using the same Customer Address API can be assigned the Sapphire API plan.

API Plans enable more granular rate limit and quota control at the application API key level in addition to the limits set at the Organization or API levels. And as such, you can readily define different API plans for the same organization and API to issue different rate limit and quotas with higher limits for applications deemed critical for your enterprise.

Working with API Plans

This section describes how Portal Admins or API Owners work with API plans, which includes the following functions:

Turn On API Plans

To use API plans, a Portal Admin must turn them on.

Prerequisite: Ensure that you have met the prerequisites for managing API plans.

For more information about these prerequisites, see Manage API Plans.

WARNING

- Turning on API plans is a one-time change. You cannot turn off API plans after you have turned them on.
- API plans are not compatible with API groups; switching on API plans means that you are no longer able to use API groups.

Follow these steps:

1. From the menu bar, select **Manage**, **API Plans**.
   The **API Plans** page opens.
2. Click **Turn On API Plans Now**.

API plans are turned on.

Manage API Plans

Portal Admins and API Owners can manage API plans. They can do the following:

- Set up rate limits and quotas for the API plan.
- Assign which organizations can use the API plan.

The API Plans page displays a list of the API plans.

Add an API Plan

Follow these steps:

1. From the **API Plans** page, select **Add API Plan**.
   The API Plan page opens.
2. In the **Edit Details** section, complete the following, and then select **Next**:

Setting	Description
Name	Enter a name for the API plan. **Required:** Yes The name must be unique and cannot exceed 50 characters, which can co
Enable quota for this plan	Limits the number of times an API key can query the API associated with th
Enable rate limits for this plan	Limits the number of concurrent requests an API key can send requests to Setting a low value, such as 50 requests per second, can prevent your API Service attacks or poorly programmed applications. Ensure that your API p
Public Description	Enter a description that can help API Owners, Org Admins, and Developers applications.

3. In the **Add APIs** section, select **Add** to add the public or private APIs displayed in the **Available APIs** list. You can make an entry in the *Filter* search box to display a filtered list of available APIs. The APIs you add display in the **Selected APIs** list. Select **Remove** to remove the API from the API Plan. A pop-up window opens prompting you to confirm the removal.
4. Select **Next**.
5. In the **Assign Organizations** section, select **Add** to add the organizations linked to the APIs you assigned to the API Plan. You can make an entry in the *Filter* search box to display a filtered list of available organizations.The organizations you add display in the **Selected Organizations** list. Select **Remove** if you want to remove the organization from the API plan. A pop-up window opens prompting you to confirm the remove.
6. Select **Create**.

A notification displays on the **API Plans** page that you have added an API plan.

Edit an API Plan

Follow these steps:

1. From the **API Plans** page, select the API plan that you want to edit.
   The API plan opens, displaying the information for that API plan.
2. *(Optional)* In the **References** section, select **APIs**, **Applications**, or **Organizations** to display a list of the APIs, applications, and organizations assigned to the API plan.
3. Select **Edit**.
   The **Edit API Plan** page opens.
4. Update the settings as needed in the **Edit Details**, **Assign APIs**, and **Assign Organizations** sections.
5. Select **Save**.

Delete an API Plan

From the **API Plans** page, select **Delete** next to the API plan that you want to delete.

View and Choose API Plans

Org Admins and Developers can limit individual API consumption within an application by subscribing the application to an API plan. This article describes how Org Admins and Developers can:

- View the details for an API plan (such as the description, rate, and quota limits).
- Choose an API plan for visible APIs.

Permissions for Choosing an API Plan

By default, Org Admins and Developers can add APIs and choose API plans for their applications. The Org Admin can also create applications.

> **NOTE**
> If the Portal Admin or API Owner has enabled the Edit Application Request Workflow setting, the Org Admin and Developer can still edit applications but the change will be pending until the Portal Admin or API Owner approves the change.

Choose an API Plan for an Application

Follow these steps:

1. From the menu bar, select **Manage**, **APIs**.
   The **Applications** page opens.
2. From the **Actions** drop-down for the application that you want to edit, select **Edit**.
   The **Details** page opens.
3. Select the **API Management** tab.
   A list of API plans is displayed on the **API Management** page. The details for each API plan (Description, Quota, and Rate Limit) are provided so you can make an informed decision when choosing the API plan.
4. Select the API plan to which you want to subscribe the application.

Add an API Plan to an Application

Follow these steps:

1. From the menu bar, select **Manage**, **APIs**.

The APIs page opens.
2. Select the API for the organization.
3. Select **Actions**, **Add API to Application**. A list of available applications to which you can add the API are displayed.
4. Select the application.
   The **API Management** page opens for that application, listing the API plans. The details for each API plan (Description, Quota, and Rate Limit) are provided.
5. When the EULA pop-up window opens, accept the terms and conditions. The **Select an API Plan** pop-up window opens, displaying the details for each API plan.
6. Select the API plan, and then select **Confirm**. The App details page opens where you can view the newly added API as a tile and the API plan information on the API tile.

Manage API Groups

> **WARNING**
>
> If you have turned on API plans, you will no longer be able to use API groups.

As a Portal Admin or API Owner, you can add, edit, deprecate, and delete API groups. API groups are collections of APIs that Org Admins and Developers can consume. You can group your APIs across your organizational and business boundaries by adding them to API groups. Org Admins and Developers can consume API groups while adding and editing their applications.

For more information about the roles and permissions for managing API groups, see Get Started - User Types, Roles and Permissions.

> **TIP**
>
> You can also manage your API groups by way of the Portal API (PAPI) or use this API in your scripts for managing API groups.
>
> For more information about the Portal API, see Portal API (PAPI).

In this article:

Create an API Group

1. From the menu bar, select **Manage, API Groups**.
   The list of API Groups appear on the API Groups page.
2. Select **Add API Group**.
   The Add API Group page opens.
3. In the **Group Details** section of the page, complete the following fields:
   API Group Name
   The name of the API group that you are creating.
   Unique: Yes
   Maximum length: 255
   Required: Yes
   Description
   The description of the API group that you are creating.
   Maximum length: 255
   Required: No
4. Add and remove APIs to your API group. In the **APIs** section of the page, the **Available APIs** section lists the APIs that you can add to your API group. You can only add enabled APIs to your API group. To add an API to your API group, next to the API that you want to add to your API group, select **Add**.

NOTE

Because this API group is new and you have not added APIs to your API group yet, the Selected APIs section is empty.

5. Select **Save**.

Your API group is created. See your group in the list on the **API Groups** page. The API group is enabled by default.

View your List of API Groups

You can view your list of API groups on the API Groups page. The following information is displayed:

- The state of each API group
- The number of applications that have added the API group
- The number of organizations that are associated to the API group

Edit an API Group

You can edit your API groups, including those API groups an Org Admin or Developer has added to their application.

You can edit your API group in the following ways:

- Add APIs to your API group.
- Remove APIs from your API group.
- Change the API group name.
- Change the API group state.

Add and Remove APIs from an API Group

You can add public and private APIs to your API group. You can remove only APIs that an Org Admin or Developer has not already added to their applications by way of your API group.

Follow these steps:

1. From the API Groups page, on the Actions menu next to the API group you want to delete, select **Edit**.
   The Edit API Group page appears.

   NOTE

   If you already added an API to your API group and the state of the API changes from enabled to disabled, the following message appears at the top of the **Edit API Group** page:

   ```
   This API Group contains disabled APIs and will not be available for applications of same
    organizations.
   ```

   Applications within your organization can only use API groups that include enabled APIs. Applications within all other organizations can use API groups that include disabled APIs.

   Best Practice: Remove disabled APIs from your API group.
2. Do the following:
   - **Add APIs to the API group:** In the APIs section of the page, in the **Available APIs** section, next to the API that you want to add to your API group, select **Add**. Only enabled APIs are listed.
     The API is added to the **Selected APIs** section.
   - **Remove APIs from the API group:** In the APIs section of the page, in the **Selected APIs** section, next to the API that you want to remove, select **Remove**.
     The API is removed from the **Selected APIs** section.
3. Select **Save**.

Your changes to the API group are saved.

Edit the State on an API Group

You can change the state of your API groups to enabled or deprecated. From the Edit API Group page, in the Group Details section of the page, select **Enabled** or **Deprecated**, and then select **Save**.

Delete an API Group

Prerequisite: An Org Admin or Developer has not added the API group to an application (the application is not using the API group).

1. From the API Groups page, on the Actions menu next to the API group you want to delete, select **Delete**.
2. Select **Ok** to confirm the deletion of the API group.

The API group is deleted and no longer displays on the API Groups page.

Manage Policy Templates

This section provides information about how to manage, publish, and deploy policy templates in API Management SaaS.

About Policy Templates

Publishers use policy templates to customize how a policy or policy fragment on the API proxy processes calls to an API.

The following methods are available for adding or importing policy templates into API Management SaaS:

Method	Description	Use Case
Proxy enrollment	Policy templates are added to API Management SaaS as part of the proxy enrollment and during subsequent updates to the integration software.	Also known as pre-baked policy templates, this method adds standard authentication, quota, and rate limit policy templates to API Management SaaS. For more information about proxy enrollment, see Integrate On-Premise API Proxies.
Policy Manager	Use the Policy Manager to convert a policy fragment into an encapsulated assertion, then enable the assertion so that API Management SaaS can discover and publish it.	API proxy administrators enable individual assertions and policies to be published through API Management SaaS. For more information, see Add Policy Template using Policy Manager.
Gateway bundles	Policy authors create policy bundles in the Gateway Policy Plugin. The bundle can include assertions as well as other policy- or environment-specific entities.	Portal Admins can import Gateway bundles into API Management SaaS, centrally manage the assertions, policies, and/or services contained with them, and deploy them to the proxies managed in API Management SaaS. For more information, see Manage Policy with Gateway Bundles.

NOTE
On upgrade to the portal integration bundles at Gateway, policy template synchronization process occur through portal-data. Once the portal integration bundles are upgraded, the custom policy templates created at Tenant Enrolled Gateways no longer use PSSG for synchronization to portal.

Review the following information:

Add Policy Templates using Policy Manager

API proxy administrators can use the Policy Manager to convert a policy fragment into an encapsulated assertion and add it to the API proxy. Once converted, they can allow API Management SaaS to discover and publish the assertion, at which point it becomes a "policy template".

The API proxy administrator can add input and output arguments to the encapsulated assertion. For information about the encapsulated assertions, see the topic "Encapsulated Assertions" in the online documentation for the Layer7 API Gateway.

IMPORTANT

- We strongly recommend that only API proxy administrators with experience developing policies on the API Gateway create policy templates.
- To create modular policy templates that API publishers can combine is challenging, so keep them simple.
- Policy templates are available only to APIs published on API Portal.
- Each template needs at least one input argument that is displayed in API Portal.
- Ensure that you migrate non-default policy templates across all the enrolled proxies using GMU to avoid policy template deletion during the respective policy template sync.

Follow these steps:

1. In the Policy Manager, connect to the API proxy.
2. On the Tasks menu, select **Create Policy**.
3. In the Policy Properties dialog, enter a name and select **OK**. The policy fragment appears on the Policy Manager Services and Policies list and Policy Development window.
4. Open the Internal Assertions palette, and then drag the **Set as Portal Publishable Fragment** to the policy fragment in the Policy Development window.
5. In the Policy Development window, construct the policy.
6. In the Services and Policies list, right-click the policy fragment, and then select **Create Encapsulated Assertion**.
7. Confirm the auto-population of inputs and outputs. The Encapsulated Assertion Configuration Properties dialog opens.
8. Enter a template name and a description.
9. On the **Palette Folder** menu, select a palette.

   NOTE
   API Portal displays the name and description. Ensure that the description clearly describes to API publishers how they can use the template alone or with other templates.
10. Add input and output arguments to the encapsulated assertion. Do not include periods in the names of input arguments, such as "test.email".
    To display an input argument, assign the string, integer, decimal, or Boolean type to the argument, and then select the checkbox next to the **Show in assertion properties dialog**.
    To make a displayed input argument mandatory, add an asterisk to the end of the input argument label.
11. Select **OK**. The dialog closes.
12. Select **Save and Activate**.

Verify That a Policy Template is in API Portal

Follow these steps:

1. Log in to API Portal as a Portal Admin.
2. From the menu bar, select **Manage**, **APIs**.
3. Select **Add API**, or select an existing API, and then select **Actions**, **Edit API Details**.
   The **Details** page appears.
4. Enter or confirm the API details, then select **Save & Next**. Alternatively, you can select the **Policy Templates** tab.

A list of applied policy templates display on the **Policy Templates** page, as well as the Select Policy Template dropdown.
5. Click the dropdown to verify that the added policy template is available.
6. Click the **+** icon to add the policy template, and expand it to verify that the template displays any fields that were configured to appear.

Edit a Policy Template in Policy Manager

Follow these steps:

1. In the Policy Manager, connect to the API proxy.
2. Edit the template policy:
   a. Open the policy in the Policy Development window.
   b. Edit the policy.
   c. Select **Save and Activate**.
3. Edit the template's encapsulated assertion:
   a. Open the encapsulated assertion in the Encapsulated Assertions Property dialog.
   b. *Optional:* Edit the name and description of the encapsulated assertion.
   c. *Optional:* Add, edit, or delete arguments.

      NOTE
      Do not include periods in the names of input arguments, such as "test.email".

Delete A Policy Template in Policy Manager

Follow these steps:

1. In the Policy Manager, connect to the API proxy.
2. On the Tasks menu, select **Manage Encapsulated Assertions**. The Manage Encapsulated Assertion Configurations dialog opens.
3. Select the template's encapsulated assertion, and then select **Remove**. Close the dialog.
4. In the Policy Manager's Services and Policies list, right-click the policy fragment, and then select **Delete Policy**.
5. Use the preceding policy template verification procedure to confirm that the deleted template does not appear in API Portal.

Manage Policy with Gateway Bundles

This page describes two kinds of Gateway bundles: system Gateway bundles and standard 'non-system' Gateway bundles.

In this section:

About System Gateway Bundles

A system Gateway Bundle is a policy bundle that is included in API Developer Portal. System bundles are given a “l7.apim.system” namespace prefix and tagged as “System”. They cannot be deleted or removed.

As a Portal admin, you deploy system Gateway Bundles to the proxies managed in Layer7 API Developer Portal when you want to enable Portal features that require the system bundle.

For example, in order to enable the option to define rate limits and quotas at the API assignment level, the “l7.apim.system - Rate & Quota Policy Template” system bundle must be deployed to all proxies. After its deployment, you may define a rate limit and quota with an API assignment level for Portal-published APIs.

To learn how to

- Deploy system bundles, see Upload and Deploy Gateway Bundles.
- Publish an API using the Rate and Quota Policy template, see Configure Rate Limit and Quota During API Publishing.

About Gateway Bundles

A Gateway Bundle is a set of files that the Gateway Policy Plugin generates to simplify the management of policies and expose policy- or environment-specific entities.

Policy authors can create policy bundles in the Gateway Policy Plugin and incorporate them into the CI/CD pipeline for easier upgrades and migrations. As a Portal Admin, you can import Gateway bundles into API Management SaaS, centrally manage the assertions, policies, and/or services contained with them, and deploy them to the proxies managed in API Management SaaS.

Each Gateway bundle consists of three files:

- **<name>-<versionmajor>.<versionminor>.<versionbuild>.metadata.yml**: Includes information about the generated bundle along with its dependencies.
- **<name>-<versionmajor>.<versionminor>.<versionbuild>.install.bundle**: Packages policies, encapsulated assertions (encass), services, and other Gateway entities.
- **<name>-<versionmajor>.<versionminor>.<versionbuild>.delete.bundle**: Enables undeployment of the bundle.

Examples:

- `sample-1.0.00.metadata.yml`
- `sample-1.0.00.install.bundle`
- `sample-1.0.00.delete.bundle`
- `sample-1.0.00-full.install.bundl`

There are four types of Gateway bundles:

- **Encapsulated Assertion:** Contains an encapsulated assertion (encass) along with its policy dependencies. Can be reusable if the `l7template` attribute is true.
- **Policy:** Contains a policy fragment along with its dependencies.
- **Service:** Contains a service (Web API or SOAP) definition along with its service policy dependencies.
- **All:** Contains all of the above components.

> **NOTE**
> API Management SaaS supports all Gateway bundle types except Service.

For more information about bundles built in the Gateway Policy Plugin, including the attributes within the metadata file, see the Gateway Policy Plugin documentation.

How Gateway Bundles Work

Gateway bundles enable you to export any code, configuration, or environment from the API Gateway through the Gateway Policy Plugin. After the Gateway bundle is uploaded into API Management SaaS, you can manage and deploy the bundles to enrolled proxies. This leverages API Management SaaS as the single source of management console for policy lifecycle management, in addition to lifecycle management of APIs, applications, proxies, and other entities.

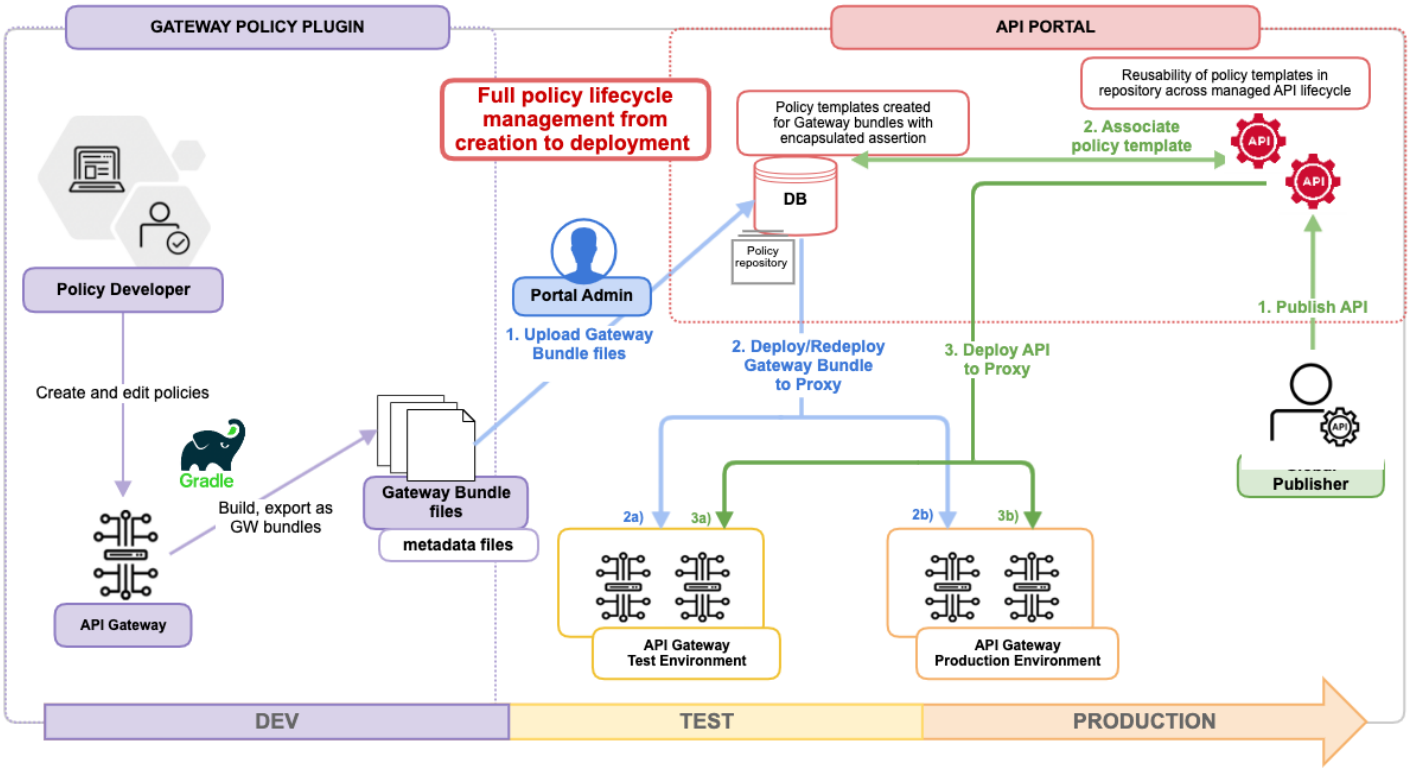
For Gateway bundles that have encapsulated assertions (encass), API Management SaaS creates policy templates from the associated metadata. The policy templates are available as reusable policies to API publishers while publishing APIs. Portal Admins can also deploy or promote policy changes through dev, staging, and production environment proxies managed in API Management SaaS.

The `metadata.yml` file of a Gateway bundle contains basic details such as group name, name, version, tags, and defined entities, environment dependencies, and other dependencies. API Management SaaS uses the metadata to generate a policy template that can be managed and reused in APIs published through API Management SaaS.

After the Gateway bundle is uploaded into API Management SaaS, API Management SaaS acts as the single source of truth for policy management. Heterogenous policy deployment across environments are possible as API Management SaaS acts as the control plane for deployment of policies to all the proxies. This also helps with scalability by enabling API Management in multi-geographic, multi-cluster deployments.

Policy Lifecycle Management Overview

The following diagram describes an overview of policy lifecycle management and how Policy Developers, Portal Admins, and Publishers (API Owners, Org Publishers) collaborate across Layer7 products when using Gateway bundles.



Policy management with Gateway bundles includes the following workflows:

- **Policy authoring:** Includes creation and management of policies in the API Gateway, and creation and export of Gateway bundles using the Gateway Policy Plugin. See the Gateway Policy Plugin documentation for more information on creating Gateway bundles.
- **Bundle management:** Includes upload of Gateway bundles to API Management SaaS, and federated bundle deployment using API Management SaaS.

  NOTE
  You can deploy bundles only on demand.
- **Policy template management:** Includes API publishing and association of policy templates through API Management SaaS.

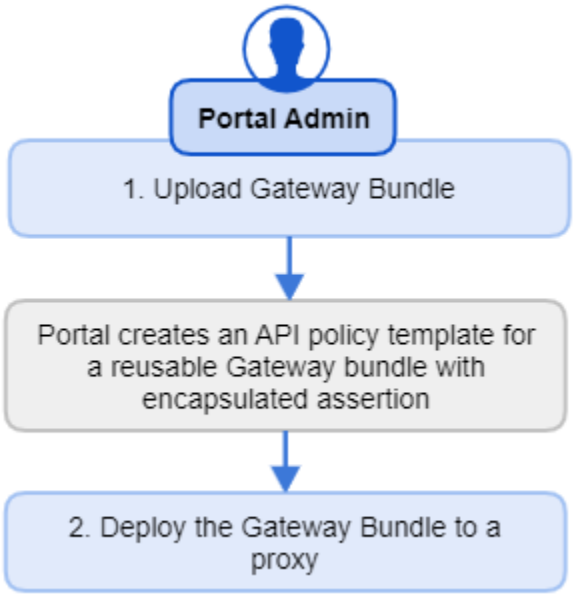
  TIP
  Next steps:

  - Upload and Deploy Gateway Bundles
  - Associate and Deploy Policy Templates

Upload and Deploy Gateway Bundles

You can upload and deploy Gateway bundles as part of the bundle management workflow. You can also undeploy, redeploy, delete, view details for, and update your Gateway bundles.

> **NOTE**
> Typically, this workflow is intended for Portal Admins.



Upload a Gateway Bundle to API Portal

Prerequisites:

Ensure that you have the metadata, bundle, and delete bundle files ready. Maximum size per file is 15 MB.

Follow these steps:

1. Log in to API Management SaaS as Portal Admin.
2. From the menu bar, select **Manage**, **Gateway Bundles**.
3. Select **Add Bundle**.
4. Drag and drop the files into the uploader window, or click the uploader window and select and upload your files.
5. Select **Save**. The files are uploaded and are listed on the page.

> **TIP**
>
> - The bundle name must be unique. Do not change the bundle file name. The combination of group name, name, and version must be unique.
> - You can update a Gateway bundle by uploading a new build of the bundle to API Management SaaS. For more information, see Update a Gateway Bundle.
> - You cannot import the same policy twice. If encapsulated assertions with the same attributes already exist in API Management SaaS, an error message is shown.

NOTE

- For bundles of the encapsulated assertion type (the **l7template** field value must equal "true" in the metadata file), API Management SaaS creates a policy template so that different APIs can reuse the policy.
- API Management SaaS does not support the Service bundle type, and API Management SaaS rejects the upload of this bundle type.

Deploy a Gateway Bundle to a Proxy

Follow these steps:

1. From the **Gateway Bundles** page, select the bundle that you want to deploy.
2. Select the **Gateway Bundle Deployments** tab. The cards show the deployment status of the bundle on all proxies.
3. Select the proxy for which you would like to deploy the Gateway bundle, and then select **Deploy**.

Undeploy and Redeploy a Gateway Bundle

Prerequisites:

Before undeploying a bundle from a proxy, ensure that:

- No API referencing the bundle's policy template is currently deployed, or
- Any API recently detached from the policy template has been redeployed, i.e. no legacy policy template reference remains on the proxy.

The following scenarios will result in an error:

- The bundle is undeployed while a deployed API is still associated with the policy template. The API throws an error when requests are made to this API.
- An API is detached from the policy template, but has not been redeployed to the specific on-demand proxy. Although you are able to undeploy the bundle, the proxy will throw an error.

Follow these steps:

1. From the **Gateway Bundles** page, select the bundle that you want to undeploy or redeploy.
2. Select the **Gateway Bundle Deployments** tab. The cards show the deployment status of the bundle on all proxies.
3. Select the proxy for which you would like to undeploy or redeploy the Gateway bundle, and then select **Undeploy** or **Redeploy**.

Delete a Gateway Bundle

Prerequisites:

- The bundle is not deployed to a proxy. If it is, undeploy the bundle from all proxies.
- No API is referencing the bundle's policy template. If it is, disassociate the API from the policy template. For more information, see Associate and Deploy Policy Templates.

Follow these steps:

1. From the **Gateway Bundles** page, select the bundle that you want to delete.
2. Select **Actions**, **Delete Bundle**.

View the Gateway Bundle Details

As a Portal Admin, you can view the list of Gateway bundles that have been imported into API Management SaaS, along with these details:

Metadata

View the bundle's details by selecting the **Metadata** tab. Details include bundle type, extracted policy template, APIs consuming the bundle, date of creation, group name, and bundle description.

> **TIP**
> You can download and view the bundle's metadata file to resolve any possible conflict of subsequent bundle version uploads. To download the metadata file, go to **Metadata > Assets > Download**.

Deployments

View bundle deployment details by selecting the **Gateway Bundle Deployments** tab. The cards show the last updated time (when the bundle was deployed), the bundle version and build that is currently deployed, and the deployment state of the bundle:

- **Deployed** - the bundle is deployed to the proxy.
- **Not Deployed** - the bundle has not been deployed to the proxy.
- **Pending Deployment** - the deployment of the bundle to the proxy is pending.
- **Pending Undeployment** - the undeployment of the bundle to the proxy is pending.

> **TIP**
> Click **More Info** to view deployment errors. The deployment shows the proxy's Restman service response.

APIs

View the list of APIs referencing the bundle by selecting the **APIs** tab. This is particularly useful before undeploying or redeploying a bundle. You can filter APIs referencing the bundle by API name, the proxies to which it is deployed, and the API state.

> **NOTE**
> The APIs tab only shows for a Gateway bundle of the Encapsulated Assertion type.

Update a Gateway Bundle

You can update a Gateway bundle version by uploading a new build of the bundle to API Management SaaS.

Prerequisites:

- The bundle is of the encapsulated assertion type and the l7template field value must equal to **"true"** in the metadata file.
- The <versionbuild> in the filename is different from the previous version.
- The <versionmajor> and <versionminor> in the filename are identical to the previous version. See Manage Policy with Gateway Bundles for more information on bundle naming standards.
- The groupName, name, and moduleName are identical to the previous version.

The following restrictions apply:

- If you upload a new version instead of a new build of a Gateway bundle, it is treated as a different bundle and an in-place update will not be possible.
- Uploading a new build of a bundle’s version overwrites the older build, but does not do so automatically on the proxy. The older build will still be deployed until you choose to perform bundle update on your selected proxy.
- Updating a bundle to the latest build on a proxy automatically redeploys the bundle on the selected proxy. This process cannot be reverted.
- Upon redeployment, APIs referencing the policy template associated with the bundle will be automatically updated.

Follow these steps:

1. Upload a new build of your specified Gateway bundle version. For upload instructions, see Upload a Gateway Bundle to API Portal.

   > **NOTE**
   > If the prerequisites are not met, a validation error will appear.

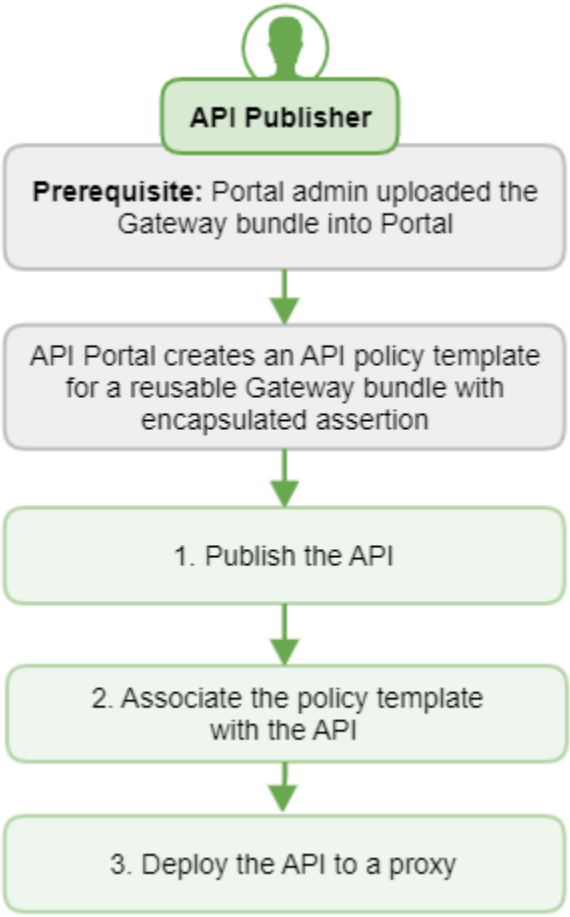
2. From the list of Gateway bundles, click the bundle that you want to update.
3. Select the **Gateway Bundle Deployments** tab. The cards show the registered proxies. If a proxy is still using an older build of the bundle, an *Update* option appears in the proxy card, along with version information.
4. Click **Update**. Confirm your selection when prompted.

   TIP
   Click **More Info** to view any deployment error. The deployment shows the proxy's Restman service response.

Associate Policy Templates to API

You can associate policy templates to an API and deploy the API to proxy as part of the API publishing workflow.

NOTE
Typically, this workflow is intended for the API Publisher.



Prerequisites:

The Gateway bundle is deployed to the proxy that you intend to deploy your API to.

NOTE
If the Gateway bundle for the selected policy template is not deployed to the proxy, the API deployment will fail.

Associate a Policy Template to an API

Follow these steps:

1. Log in to API Portal as Publisher.
2. From the menu bar, select **Manage, APIs**.
3. Choose from the following options:
   - If you are publishing and deploying an API for the first time, click **Add API** and, as you add your API details, select **Policy Templates** from the left-frame menu.
   - For already published APIs, select the API that will consume the policy, go to **Actions > Edit API Details**, and select **Policy Templates** from the left-frame menu.
4. Select the desired policy template in the drop-down list. You can select more than one policy template.
5. Add the policy template to the API so the policy will be applied when requests to this API are processed.

   NOTE
   Policy templates are formatted to show the bundle's group name, name, and version. For example: BundleGroupA - BundleNameB - 1.0.0.
6. Click **Save & Next**. The policy is now associated with the API.
7. Continue with the workflow to update the API.

Deploy an API to a Proxy

Prerequisites:

The Gateway bundle has been deployed to the proxy where the API will be deployed.

NOTE
If the Gateway bundle for the selected policy template is not deployed to the proxy, the API deployment will fail.

For instructions on deploying an API, see Deploy APIs.

Monitor

Analytics help you understand common scenarios and make better business decisions:

- Which APIs have the most hits?
- How API traffic is trending over a period of time?
- What is the average API response time?
- What is the error rate?
- What is the average latency time?
- What are the usage limits of my Account Plan?

Analytics data is retained for **two** years. You can enable or disable analytics while deploying the Portal stack. The **Monitor** menu appears on the menu bar if you have enabled the analytics stack.

When enabled, you can monitor and visualize traffic, latency, error rate, and usage trends of APIs by navigating to the corresponding reports. You can filter data based on APIs and/or applications and/or organizations, and select a specific time period as well. These reports generate related graphs based on the filters that you select:

- **Traffic, Latency & Errors Report:** This report shows traffic, response latency, and error rate trends of APIs over the selected period of time. You can compare current metrics with historical data and understand the corresponding trends for the selected period.
- **Quota Consumption Report:** This report shows the monthly quota consumption and the daily quota consumption of APIs and applications over a period of time as determined by the assigned account plan.
- **Custom Report**: Use this report to drill-down to specific API metrics and visualize the exact data based on your business requirements.

Portal Admins and API Owners can access metrics for all the APIs, applications, and organizations across the tenant.

Org Admins and Developers can access metrics for the APIs and applications that are specific to their organization.

Org Publishers can filter data by organization and proxies. To understand the consumption of their APIs across organizations, they can access metrics and visualize the data for all the APIs and applications owned by their organization as well as applications of other organizations using their APIs. See the individual report sections for the specific details.

Traffic, Latency and Errors Report

The **Traffic, Latency & Errors** report shows traffic, response latency, and error rate trends of APIs, organizations, and applications over a period. You can further compare current metrics with historical data and understand the corresponding trends.

From the menu bar, select **Monitor, Traffic, Latency & Errors** to access this report.

By default, all the selections you make in the filters are saved and you can view the reports with these selections the next time you access the reports.

Click **Export** at the top-right corner to export the trend charts.

Data Source Filters

Data source filters are displayed based on your access permissions. If you have access to only one entity like an API or an organization, the filter is automatically set. When you do not apply any filter, you see the overall traffic for the tenant.

You can filter the data sources based on the following entities. These filters are stackable:

- APIs
- Applications
- Organizations

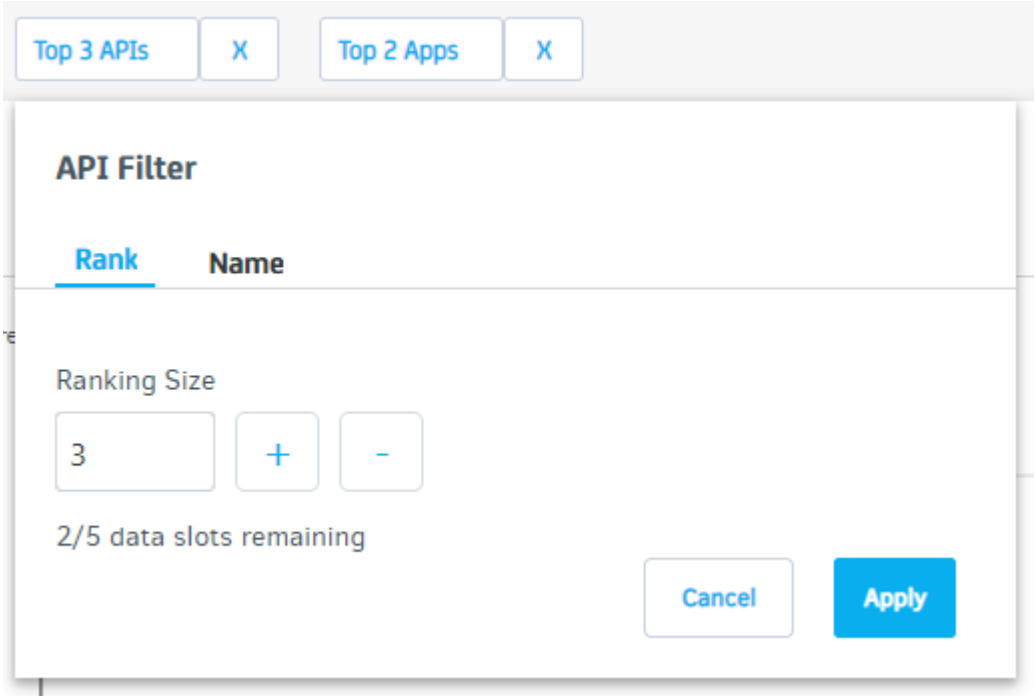
> **NOTE**
> Org Publishers can view and select any organization and filter data across all organizations of the tenant. They can also view and select applications that consume the APIs owned by their organization.

Default filter: Top 3 APIs, Last 7 Days

You can further sort each entity by either of the following:

- **Rank:** Use this option to sort the top-ranking entities.
- **Name:** Use this option to sort based on the entity name.

Click the filter to see these options as shown in the following image:



You can visualize up to **10** data slots for the entities to achieve the possible combinations. This limit is cumulative across all the selected filters. The **data slots remaining** label shows how many more entities you can choose to add.

Example: Following are some examples to understand the data source filters:

- As the filters are stackable, if you select **Top 3 APIs** and **Top 2 Apps**, it means that the report would include data for the top two applications within the top three APIs with six lines in the graph.
- If you select **3** in **Rank** for the **API Filter**, it means that the top three APIs with the highest ranking are included in the **APIs** filter.

Date Range Filters

Select a custom time period to see specific reports. Click **Edit** under the **Date Range** section to see the options. You can select current timelines, select a time interval starting now and going back by the stated number of days, or select a date range. You can select a maximum of five years in the date range filter.

Default filter: Last 7 days

Data is rolled up and plotted in the graph based on the following time intervals:

- **Last 1 Hour**: (if supported) The data is plotted by minute.
- **Last 24 Hours**: (if supported) The data is plotted by hour.
- **Last 7 Days**: The data is plotted by day.
- **Custom Date Range:**
  - Up to 2 days, the data is plotted by hour.
  - 3 days to 60 days, the data is plotted by day.
  - 61 days to 1 year, the data is plotted by week.
  - Beyond 1 year, the data is plotted by month.

Graphs and Metrics

Each report panel has a graph and a Metrics section, which shows the data. You can view the details corresponding to any data point by hovering over the data point in the graph; all the relevant details are included in the pop-up:

- **X-axis:** Displays the date and timezone based on your selected date range.
- **Y-axis:** Displays X interval margin around the minimum/maximum of the lines that are plotted.

The Metrics section in the Traffic, Latency, and Errors report shows the aggregated data based on the filters that you selected. Click a metric to add or remove the corresponding line from the graph. The subtitle text under the **Metrics** heading shows the number of lines on the graph and the overall limit.

The key metric is the first datapoint to be listed in the Metrics section. The other datapoints are historical datapoints that you can use to compare with the current metrics. The key metric is always visible if one of the historical lines is toggled on.

Traffic

This graph shows the traffic trends of the selected APIs, applications, and organizations in the selected time period:

- **Key Metric:** Total API Hits
- **Historical Data:** 30 Days Ago, 52 Weeks Ago

Latency

This graph shows the average, minimum, and maximum latency trends of the selected APIs, applications, and organizations in the selected time period. Latency is plotted only for successful API requests:

- **Key Metric:** Average Latency
- **Other Metrics:** Maximum Latency, Minimum Latency
- **Historical Data:** 30 Days Ago, 52 Weeks Ago

Errors

This graph shows the error percentage for the API hits of the selected APIs, applications, and organizations in the selected time period. All API requests that return HTTP error responses are categorized as errors:

- **Key Metric:** Error Rate
- **Historical Data:** 30 Days Ago, 52 Weeks Ago

Quota Consumption Report

The **Quota Consumption** report shows the monthly quota consumption, the daily quota consumption, and the hourly consumption of organizations over a period as determined by the assigned Rate Limit and Quota. You can further compare current metrics with historical data and understand the corresponding trends.

From the menu bar, select **Monitor, Quota Consumption** to access this report.

By default, all the selections you make in the filters are saved and you can view the reports with these selections the next time you access the reports.

Click **Export** on the top-right corner to export the trend charts.

> **NOTE**
> For more information about account plans, see Manage API Usage.

In the **Daily Quota Consumption** graph, the consumption at any point in time is calculated using the number of hits in the last 24 hours. Similarly, in the **Monthly Quota Consumption** graph, the consumption at any point in time is calculated using the number of hits in the last 30 days. The quota usage plotted is not static; it is rolled over at the end of the day for the Monthly Quota Consumption graph and at the end of the hour for Daily Quota Consumption graph.

The quota consumption that is plotted at any data point on the x-axis is the actual usage at that point in time.

> **NOTE**
>
> The default quota level is API, and the date range is set to the Last 7 Days. To view the Hourly Quota Consumption Report, ensure that the Date Range is set to Last 24 Hours.

Data Sources and Filters

Data Sources and Filters are displayed based on your access permissions. You cannot remove the Organization filter in this report but you can sort each data source by either **Rank** or **Name**. If you have access to only one entity like an API or an organization, the filter is automatically set.

Default filter: Top 3 Organizations, Last 7 Days

> **NOTE**
> To understand the common filters like graphs, metrics, and date range, see the corresponding sections in the Traffic, Latency & Errors Report.

Monthly Quota Consumption

This graph shows the monthly quota consumption for the selected Organization, API, and API per Organization as determined by the assigned Rate Limit and Quota. Organizations that are linked to Rate Limit and Quota with monthly quota intervals are listed here.

To see the actual usage, move the cursor over the line on the graph. You can see the percentage along with the actual usage versus the monthly limit for the Rate Limit and Quota.

Daily Quota Consumption

This graph shows the daily quota consumption for the selected Organization, API, and API per Organization as determined by the assigned Rate Limit and Quota. Organizations that are linked to Rate Limit and Quota with daily quota intervals are listed here.

To see the actual usage, move the cursor over the line on the graph. You can see the percentage along with the actual usage versus the daily limit for the Rate Limit and Quota.

Hourly Quota Consumption

This graph shows the hourly quota consumption for the selected API and API per Organization as determined by the assigned Rate Limit and Quota. Set the Date Range option to Last 24 Hours to view the Hourly report at the API or API per Organization quota levels. It lists all the API and API per Organizations linked to Rate Limit and Quota with hourly quota intervals.

To see the actual usage, move the cursor over the line on the graph. You can see the percentage along with the actual usage versus the hourly limit for the Rate Limit and Quota.

Known Issues

Data shown on the Daily Quota Consumption chart does not include the current day data (DE436220)

The Daily Quota Consumption chart's X-axis and tooltip show the data for month and date but does not show the time. If you select the current day data hits, the chart shows "NO DATA".

Custom Report

You can drill down to specific API metrics and visualize the exact data based on your business requirements using custom reports. Select the metrics, group if needed, and limit the data based on your criteria using filters.

Create and visualize custom charts as panels. You can create up to three panels. Each panel includes the following fields, which you can customize:

- Title and Description
- Metric
- Dimensional filters
- Time filters

You can group or ungroup each dimensional filter. Each grouped dimensional filter only filters the data based on your selections. However, each ungrouped dimensional filter is represented on the chart as a line.

After you save the custom reports, API Portal retains the report information upon browser refresh or subsequent login. You can modify the charts later as needed.

You can retrieve the insights that are visualized in API Portal using the Metrics Query API. For more information about this API, see Metrics Query API.

From the menu bar, select **Monitor, Custom Report** to access this report.

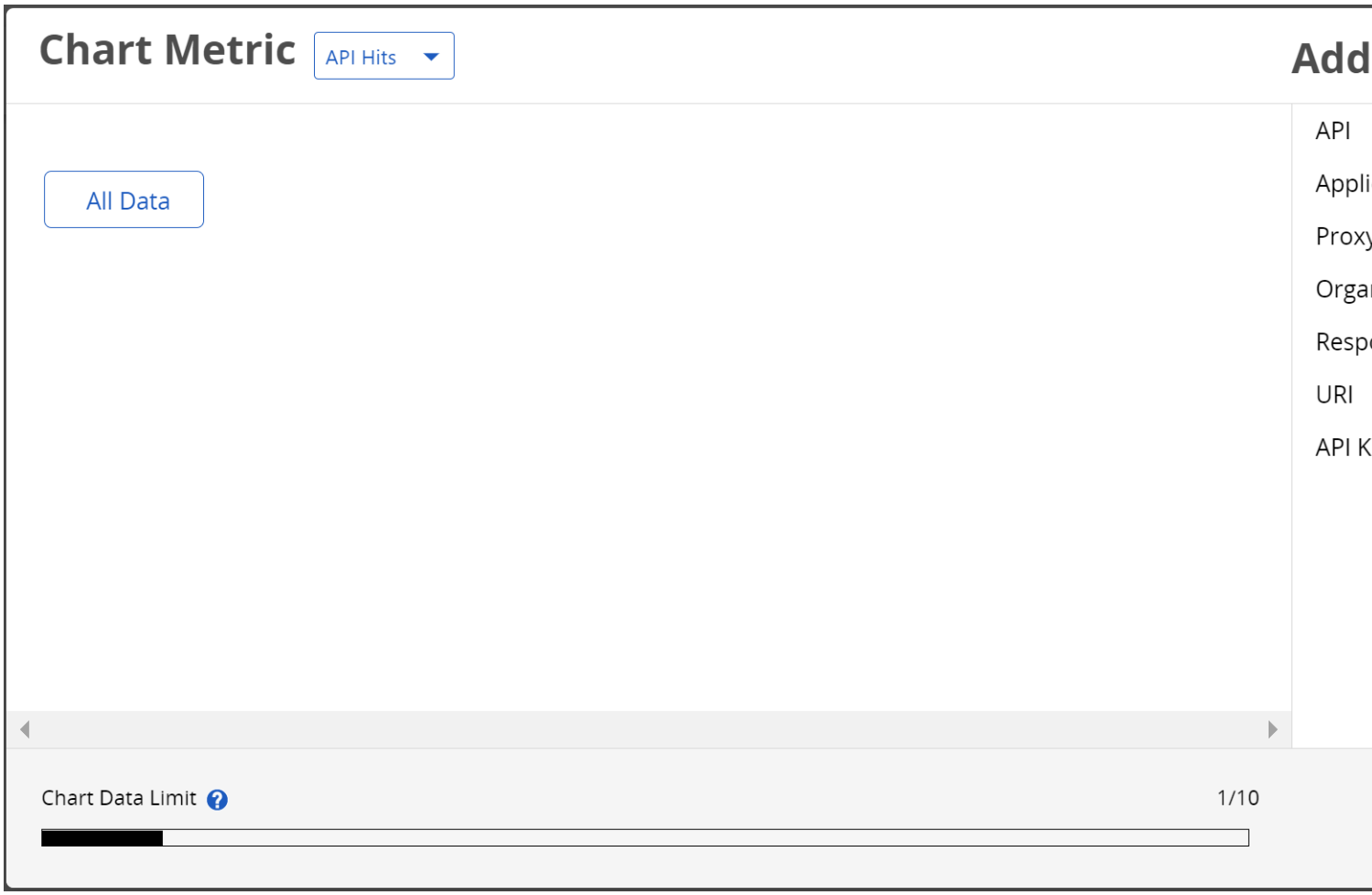
Select **Export** on the top-right corner to export the custom charts.

Create a Panel on Custom Report

For example, you can customize your analytics reports to filter or group API hits by API key.

Follow these steps:

1. Click **Add Panel** on the **Custom Report** page.
   The Chart Metric panel opens.



2. Select the chart metric from the drop-down. By default, the **API Hits** chart metric is selected.
3. Select the data source for which you want to create the chart, and then select **Apply**.

   NOTE
   Click **All Data** to fetch all the data.

   You can select from the following dimensional filters on the right-side of the panel to further drill-down data. Selecting any dimension from the right-side panel reflects visually on the left-side of the panel. Each line on the left-side of the panel represents the data flow of a line on the final chart.

   – **API:** Select APIs by Name or Rank.
   – **Application:** Select Applications by Name or Rank.

     NOTE
     The applications that are displayed are those that consume the APIs owned by the Org Publisher's organization.
   – **Proxy:** Select Gateway Server by Name or Rank.

     NOTE
     The proxies that are displayed are those that are visible to the Org Publisher's organization.
   – **Organization:** Select Organizations by Name or Rank.

NOTE
The organizations that are displayed are those that consume the APIs owned by the Org Publisher's organization.

– **Response Code:** Select Response Code by Rank, Range, or HTTP Status Code.
– **URI:** Select URI by Rank or Expression.

NOTE
If a URI pattern is a superset of any of the subsequent patterns, then the API returns the values only for the superset pattern.

- **Example:** If URI based hits are:
  - /accounts/v1 - 5
  - /accounts/v1/acc1 - 2
  - /accounts/v1/acc2 - 2
  - /accounts/v1/acc1/transactions - 3
  - /accounts/v1/acc2/transactions - 4
  - /accounts/v1/acc1/transactions/transc1 - 3
  - /accounts/v1/acc1/transactions/transc2 - 5
  - /accounts/v1/acc2/transactions/transc1 - 4
  - /accounts/v1/acc2/transactions/transc2 - 3
- Typical URI patterns that can be derived and their values:
  - /accounts/v1 - 5
  - /accounts/v1/[w] - 4
  - /accounts/v1/[w]/transactions - 7
  - /accounts/v1/acc1/transactions/[x] - 8
  - /accounts/v1/[w]/transactions/[x] - 15

– **API Key:** Select API key by Name or Rank. You can retrieve analytics such as:

- Data for an API key.
- Consumption of the API keys for an application.
- Consumption of an API from all API keys.
- Consumption of an API from the API keys for an application.
- Consumption of an API key for an application against an API.
- Consumption of all API keys for an application against an API.

NOTE
The API keys that display are those for their own organization as well as those for the applications that are managed by the Org Publisher's organization.

IMPORTANT

Analytics for Multiple API Keys (Hybrid customers only)

If you have metrics enabled (you are tracking analytics) and you are managing multiple API keys for your application, to have API Portal properly reflect and record the analytics for all API keys including the default key, update the Portal integration software on the API proxy.

For more information:

- About how to update your integration software, see Update the Integration Software on the API Proxy.
- About how to manage multiple API keys, see Manage API Keys.

–

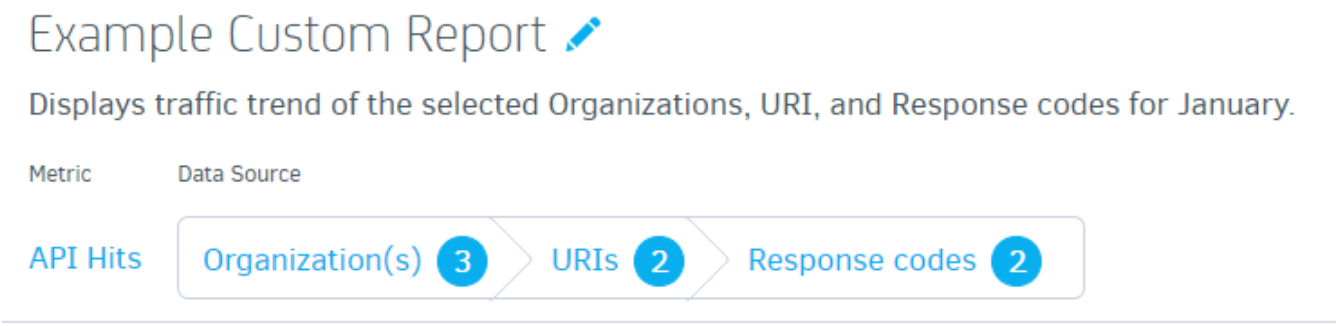
TIP

- Select the **Grouped** checkbox to group multiple entities of a data source; this also results in a single line chart for the grouped entities. For detailed information, see the following section "**Group Data on a Custom Report"**.
- Select **Remove** to delete the selected filter.

The Chart Data Limit shows the number of entities that you selected. You can select a maximum of **three ungrouped** entities.

Limit:10

4. Create the chart with the selected data source filter by selecting **Get Data**.
5. Stack filters to further drill-down the data based on your requirements. Click the data source to open the Chart Metric panel, and then select the next filter.



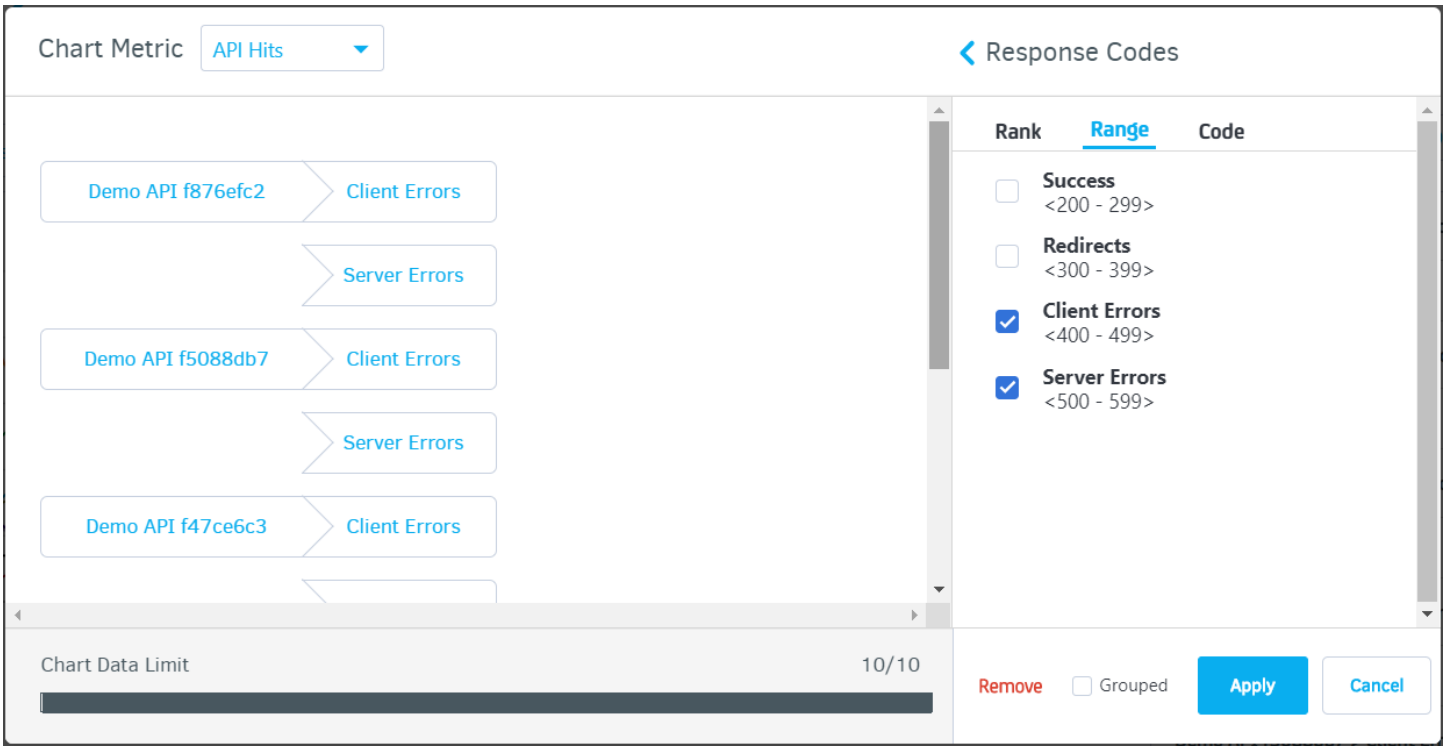
6. Click **Apply**, and then plot a chart representing data based on the stacked data sources by selecting **Get Data**. The line chart is plotted.
7. Edit the chart title and description to reflect the information that is plotted on the chart.
8. To view the data points, hover over the line chart or the legend. Details about your selections are displayed as tool-tips.
9. To edit a selection, click the corresponding data source or time range tile.

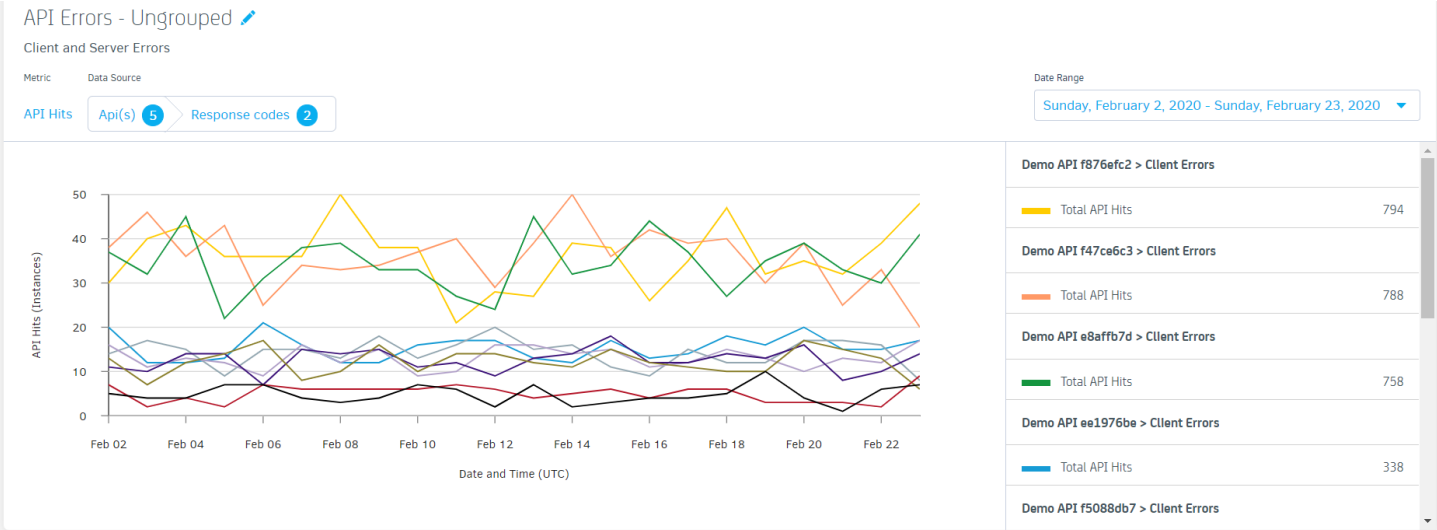
Group Data on a Custom Report

You can group multiple entities of a data source to represent it as a single line on the Custom Report line chart by selecting the **Grouped** checkbox. This helps in summarizing and visualizing large data to derive meaningful information.

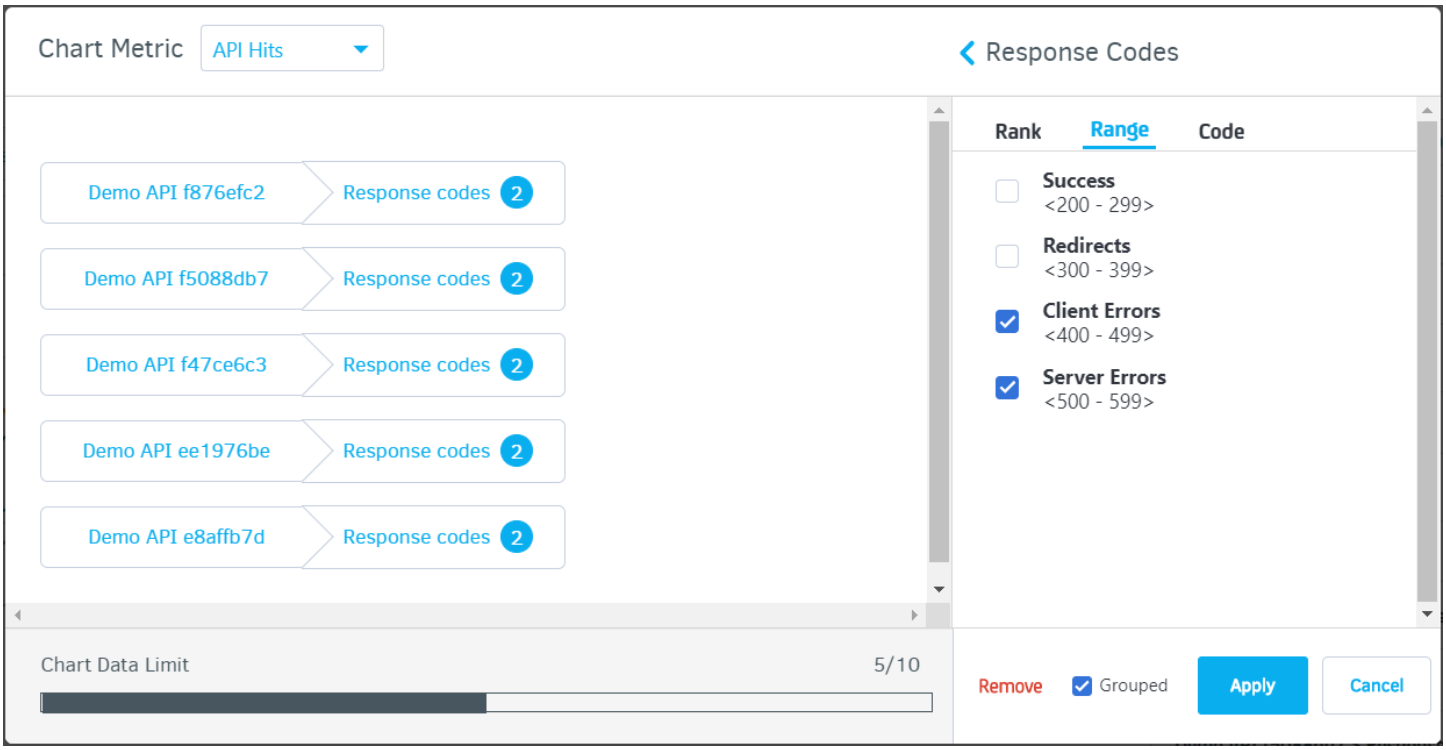
By default, the selection appears as an individual line on the chart. The number of rows you see in the Chart Metric panel is the number of lines that are plotted in the line chart. You can group data only when you have selected a data source based on Name.

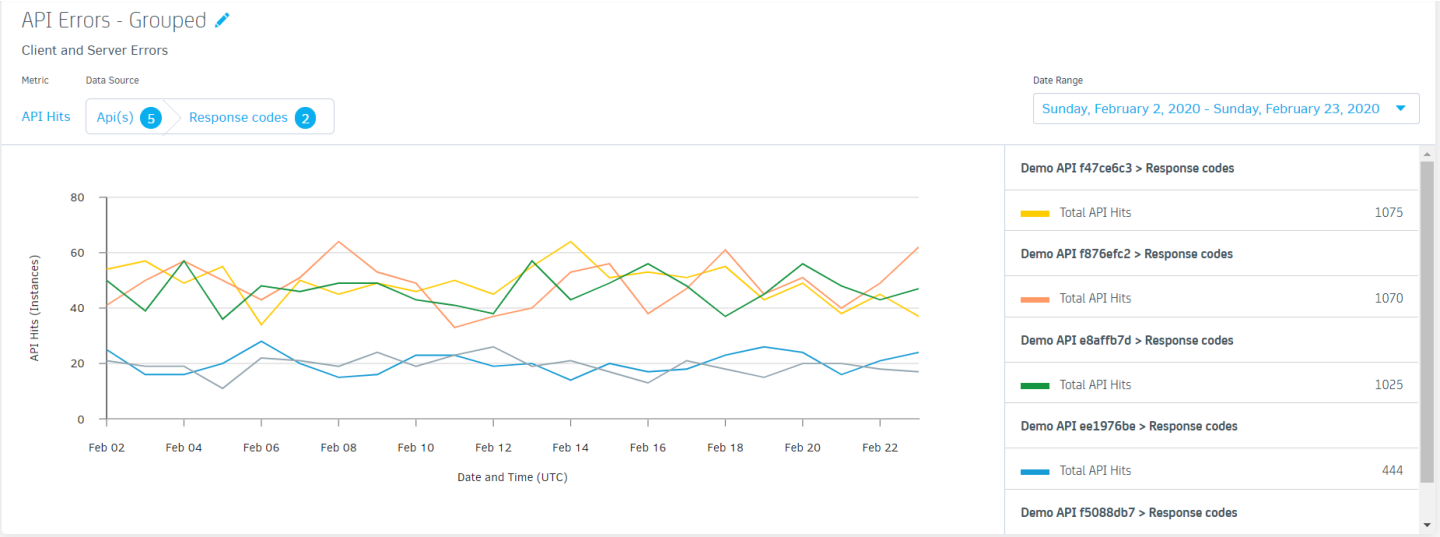
Example Custom Report without Grouped Data:





Example Custom Report with Grouped Data:





Hardware Optimizer

Use an Excel spreadsheet to calculate storage requirements. Provide input details and generate output that you can use to plan and use Analytics effectively without putting additional load that causes disrupted service.

Follow these steps:

1. Open the AnalyticsDiskUsage.xls MS Office Excel file.
2. Provide custom values for the below Input fields.

- **Ingestion Volume per day** (in transaction per day): Specify the volume based on your anticipated load or transactions per day.
- **Max Number of Applications Hits/Hour**: Specify the maximum number of anticipated hits to the application(s) in a time period of one hour.
- **Max Number of API Hits/Hour**: Specify the maximum number of anticipated APIs used in a time period of one hour.
- **Number of Gateways**: Specify the number of gateways used in your organization in a time period of one hour.

3. Press the **Enter** key to get the calculated output.

Portal APIs

In addition to the Portal UI, API Portal provides native APIs to manage your Portal, APIs, and deployments through API calls.

Contents:

- Portal API (PAPI)
- Portal Metrics API
- Login API
- Authorization API

Portal API (PAPI)

You can programmatically access key API Management SaaS entities that are exposed as RESTful resources using the Portal API (PAPI). You can call the PAPI from your external client application, or you can try it out using the API Explorer or Swagger UI in API Management SaaS. For more information about how to test with API Explorer or Swagger UI, see Test and Explore APIs.

> **NOTE**
> API Explorer is only accessible through the API Management SaaS/Ingress tenant. If you are using an external tenant, test and explore APIs using the Swagger UI instead.

In this article:

Swagger File

You can view and download the Swagger file describing the current PAPI:

Resources

The following are examples of resources that are available - for an updated list of resources, view the swagger file:

- **AccountPlans**

  > **NOTICE**
  > The /AccountPlans endpoint has been deprecated and replaced by the /api-management/1.0/rate-quotas endpoint. Use the assignmentLevel filter to filter by Organization to return rate limit and quotas (previously known as account plans).

  Use this resource to manage account plans. API publishers can use account plans to restrict cumulative usage for specific organizations.

  The AccountPlans POST method requires the `UUID` value. You can add an autogenerated UUID while creating your users using:

  ```
  "Uuid": "{{GENERATED_GUID}}"
  ```

  This method does not require values for `OldPassword` and `NewPassword` . You can leave them as null.
- **API Catalogs**
  Use this resource to retrieve public API details and various asset types associated with a public API. You may optionally bypass client authentication when retrieving by activating this feature flag: 'PUBLIC_API_CATALOG_ENABLED'.
- **ApiDeployments**

Use this resource to manage the API deployment details for a specific proxy. Proxies represent specific runtime environments and define the backed Gateways. Proxies are where APIs, applications, and account plans are deployed.

- **ApiEulas**

  NOTICE

  The /ApiEulas endpoint has been deprecated and replaced by the new api-management/1.0/eulas endpoint.

  Use this resource to manage the End User License Agreements (EULAs). EULAs are sets of legal restrictions that you can apply toward the usage of APIs managed in API Management SaaS.
- **ApiGroups**
  Use this resource to manage the API groups.
- **ApiPlans**
  Use this resource to manage API plans. Publishers use API plans to limit quota and rate limit for specific APIs.
- **Apis**
  Use this resource to manage APIs, retrieve the Swagger definition for a specific API, and retrieve relevant Gateway policy templates (using `/policyTemplates`).
- **Api Key Deployments**
  Use this resource to manage API key deployments to proxies.
- **Applications**
  Use this resource to manage applications. Applications are constructs that use one or more APIs.
- **CustomFields**
  Use this resource to manage custom fields. A custom field describes extra metadata that API Gateway administrators can use in their policies.
- **Documents**
  Use this resource to manage API documents and other custom content in API Hub such as Home page markdown content, Application Overview and Wiki documents. API documents supplement API discovery and are in addition to the Swagger API documentation. For example, an API document can include performance metrics, functional specs, best practices, and use cases. You can add markdown content to your API as documents. Wiki documents are generic documents that can be added for your API program.
- **Email Templates**
  Use this resource to retrieve and update customized Portal email notification templates. The following shows the following editable fields for a PUT update on an email template:

```
{
    "uuid": "caaaa-8a68-4dsb-4b34-846l-006348453a" ,
    "type": "EMAIL_FOR_APPLICATION_REQUEST" ,
    "subject": "Application Delete Request for ((Portal_NAME)) has been Approved" ,
    "body": "<!DOCTYPE html><html><body><h1 style=\"background-color: red; \"
            >Notification!</h1> <p>Hello:</p>....</body></html>
{
```

  NOTE

  - The “body” field accepts HTML in a single continuous line.
  - The insertion of scripts is prohibited in the email template and shall return an error indicating a potential malicious HTML code.
- **Gateway Bundle**
  Use this resource to export any code, configuration, or environment from the API Gateway through the Gateway Policy Plugin. After the Gateway bundle is uploaded, you can manage and deploy the bundles to enrolled proxies.
- **Organizations**
  Use this resource to manage organizations. Organizations are groups of one or more developers, typically representing a team or department within a business organization.
- **Proxies**

Use this resource to manage the proxies that are associated with API Management SaaS.

- **Requests**
  Use this resource to manage requests. Requests are requests from developers for acceptance or rejection by Portal Admins or API Owners.
- **Search**
  Use this resource to search for API Management SaaS entities by keyword.
- **Settings**
  Use this resource to retrieve or update an API Management SaaS setting that corresponds to a specific functionality (such as Google Analytics tracking for API Management SaaS pages and integrations).
- **Tags**

  NOTICE
  The /tags endpoint has been deprecated and replaced by the api-management/1.0/tags endpoint.

  Use this resource to manage API Management SaaS tags for APIs and Organizations.
- **Themes**
  Use this resource to manage API Management SaaS themes. Themes define the look of your API Management SaaS.
- **Users**
  Use this resource to manage API Management SaaS user accounts. You can look up supported languages for the user interface using `/languages` . Each user must have a specific developer or publisher role within an organization. You can look up available roles using `/developerRoleTypes` and `/publisherRoleTypes` , respectively.

Requests and responses are in JSON format.

For more information about JSON format, see the JSON.org website.

Access the Portal API

1. Log in to API Management SaaS as a Portal Admin for the intended tenant.
2. From the menu bar, select **Portal API**.
   Another browser tab opens from which you can access the PAPI.
3. Select the **Portal API** option from the **API** drop-down list.

The API Explorer appears in the right pane, showing the PAPI.

Authentication

PAPI calls require a valid OAuth token.

For more information about OAuth tokens, see the OAuth Community Site.

> **NOTE**
> For CA API Management OAuth Toolkit users, request the access token by posting using the `application/x-www-form-urlencoded` Content-Type.

Search and Filter API Entities

You can search for or filter API entities by adding relevant strings and values to your API call. You can also cutomize how results are shown by adding paging and sorting strings.

The following table lists supported filter options:

NOTE
Strings and values are case-sensitive. Use HTML URL encoding in place of spaces and other characters. See the following table for examples.

String	Function	Example Call
name	Filters by name where the provided string is matched as a wild card.	https://apim-ssg.example.com:9443/<tenant-id>/api-management/1.0/apis?name=pet
description	Filters by description where the provided string is matched as a wild card.	https://apim-ssg.example.com:9443/<tenant-id>/api-management/1.0/apis?description=pet%20store
apiServiceType	Filters by service type where provided value must be one of REST or SOAP.	https://apim-ssg.example.com:9443/<tenant-id>/api-management/1.0/apis?apiServiceType=REST
accessStatus	Filters by the access status of the API where provided value must be one of Public or Private. For filtering purposes, "Restricted" APIs are categorized as Private. For more information on API visibility, see Create and Set Permissions for APIs.	https://apim-ssg.example.com:9443/<tenant-id>/api-management/1.0/apis?accessStatus=Public
portalStatus	Filters by the status of the API where provided values must be one of Enabled, Disabled, Deprecated, or Incomplete.	https://apim-ssg.example.com:9443/<tenant-id>/api-management/1.0/apis?portalStatus=Enabled
orgUuid	Single UUID of Organization to filter by.	https://apim-ssg.example.com:9443/<tenant-id>/api-management/1.0/apis?orgUuid=ddc4e846-4a2d-410c-a951-218130fc6162
id	List of valid UUIDs of APIs that the caller has access to.	https://apim-ssg.example.com:9443/<tenant-id>/api-management/1.0/apis?id=8b923700-b2c0-4217-87d9-460b17f0b138,41660dec-22a3-44e1-9041-91901516440a
apiPlanUuid	Single UUID of API Plan to filter by.	https://apim-ssg.example.com:9443/<tenant-id>/api-management/1.0/apis?apiPlanUuid=e5f79052-90f4-4dd7-8ac3-9b90f0fefe22

The following table lists supported paging and sorting options:

String		
page	Specifies the page to return.	https://apim-ssg.example.com:9443/<tenant-id>/api-management/1.0/apis?page=1
size	Specifies the size of the page to return.	https://apim-ssg.example.com:9443/<tenant-id>/api-management/1.0/apis?page=1&size=24
sort	Specifies the value to sort the results, followed by either ascending (ASC) or descending (DESC).	https://apim-ssg.example.com:9443/<tenant-id>/api-management/1.0/apis?page=0&size=24&sort=name,DESC

Audit Logs

Portal Admins can see the history of actions performed on APIs within API Management SaaS by accessing audit data that are captured for APIs.

For more information about how to access audit data through a UI, see Audit Logs.

PAPI Swagger File 5.1.2

Download the sample Swagger JSON file describing the current Portal API.

For more information about the Swagger (OpenAPI) specification, see the Swagger website.

none

Portal Metrics API

Use the Portal Metrics API to retrieve various metrics for the API Portal. The API consists of the following RESTful resources:

- **errors** Retrieve error metrics for APIs or applications.
- **hits** Retrieve hit metrics for APIs or applications.
- **latency** Retrieve latency metrics for APIs.
- **usage** Retrieve usage metrics for account plans.

Requests and responses are in JSON format.

The Real-Time Analytics page in the API Portal (accessible from **Analytics** in the menu) uses this API to visualize metrics data. Integrating with the Portal Metrics API lets you retrieve metrics data for use with your own BI tools, potentially create mash-ups with other corporate data to generate a richer data set, and allows you to create your own custom data visualizations, among other potential benefits.

Access the Portal Metrics API

You can call the Portal Metrics API from your external client application, and try it out using the API Explorer.

> **NOTE**
> API Explorer is only accessible through the API Portal/Ingress tenant.

To try out the API in the API Portal:

1. Log in to the Key defintion for "aan" not found in the DITA map. as a Portal administrator for the intended tenant.
2. Select **Publish**, **Portal APIs**.
3. Select the **Portal Metrics API** option from the drop-down list. The API Explorer appears in the right pane, showing the Portal Metrics API.

Authentication

All API calls require a valid OAuth token.

Example

Download a sample Swagger JSON file from here.

alpha

Metrics Query API

You can use the Metrics Query API to retrieve various metrics for the API Portal. The Custom Report page in the API Portal uses this API to retrieve business insights that are visualized using Portal UI. Integrating with the Metrics Query API allows you to create your own custom data visualizations.

In this article, you can understand the following:

Request Fields

This API includes the following request input fields:

- **metrics**: Specifies the metrics parameters. This is a mandatory parameter.
- **timeRange**: Specifies the time related parameters
- **filter**: Specifies the filter dimensions
- **groupBy**: Specifies the group-by dimensions

Metrics

Parameter Name	Value	Default Value	
type	hits | latency	-	Specifies the type
aggregation	count | avg	-	Specifies the type

Time Range

Parameter Name	Value	Valid Values	Default Value
type	interval | period	-	period
interval (for type interval)	startDate/endDate	-	-
period (for type period)	Period	xD, xH x - integer value H - hours, D - Days	7D
aggregation	hour | day | week | month	-	all

Filter

Parameter Name	Value	Default Value	
type	and | or | selector | in | bound | regex	-	
dimension	apiId | appId | orgId | respCode | uri | gatewayServerId | apiKeyId	-	

and | or Filter

Parameter Name	Value	Default Value	
type	and | or	-	
fields	[<filter1>, <filter2>]	-	

selector Filter

Parameter Name	Value	Default Value	
type	selector	-	S
dimension	apiId | appId | orgId | respCode | uri | gatewayServerId | apiKeyId	-	S
value	<value>	-	S

in Filter

Parameter Name	Value	Default Value	
type	in	-	S
dimension	apiId | appId | orgId | respCode | uri | gatewayServerId | apiKeyId	-	S
value	[<value1>, <value2>]	-	S

bound Filter

Parameter Name	Value	Default Value	
type	range	-	Specifies the GroupBy type.
dimension	respCode	-	Specifies the dimension. This
lower	<lower>	-	Specifies the lower bound fo
upper	<upper>	-	Specifies the upper bound fo
lowerStrict	true | false	false	Specifies if strict comparison
upperStrict	true | false	false	Specifies if strict comparison

GroupBy

Parameter Name	Value	Default Value	
groupBy	[<groupBy1>, <groupBy2>]	-	Specifies the g

GroupBy Default

Parameter Name	Value	Default Value	
type	default	-	Specifies
dimension	apiId | appId | orgId | respCode | uri | gatewayServerId | apiKeyId	-	Specifies

GroupBy Top

Parameter Name	Value	Default Value	
type	top	-	Specif
dimension	apiId | appId | orgId | respCode | uri | gatewayServerId | apiKeyId	-	Specif
limit	<limit>	-	Indica

GroupBy Range

Parameter Name	Value	Default Value	
type	range	-	S
dimension	respCode	-	S
ranges	[<range1>, <range2>]	-	S

Ranges

Parameter Name	Value	Default Value	
from	<from>	-	S
to	<to>	-	S
name	<name>	%dTO%d	(

GroupBy Pattern

Parameter Name	Value	Default Value	
type	pattern	-	Specifies the type o
dimension	uri	-	Specifies the dimen
patterns	[<pattern1>, <pattern2>]	-	Specifies the list of

Patterns

Parameter Name	Value	Default Value	
value	<pattern>	-	Specifies the grouping pattern. **[d]** - indicates any number; place it a eg: /portal/api/[d] , /porta/api[d]/test **[w]** - indicates any word; place it any eg: /portal/[w]/test **[x]** - matches for anything. Place it o eg: /portal/api/[x]
name	<name>	-	(Optional) Specifies name for the gr

Example Request:

```
{
    "metrics": {
        "type": "hits|latency",
        "aggregation": "count|avg"
    },
    "timeRange": {
        "type": "interval|period",
        "interval": "1999-12-31T16:00:00/2999-12-31T16:00:00",
        "period": "numValue plus timeMetric Ex. 1D -> 1day, 24H -> 24hours",
        "aggregation": "hour|day|week|month"
    },
    "filter": {
        "type": "and|or",
```

```
    "fields": [{
            "type": "selector",
            "dimension": "appId",
            "value": ""
        },
        {
            "type": "in",
            "dimension": "apiId",
            "values": [
                ""
            ]
        },
        {
            "type": "bound",
            "dimension": "respCode",
            "lower": "400",
            "upper": "600",
            "lowerStrict": false,
            "upperStrict": true
        },
        {
            "type": "regex",
            "dimension": "uri",
            "pattern": {
                    "value":<pattern>
                    }
        }

    ]
},
"groupBy": [{
        "type": "default"
        "dimension": "apiId|appId|orgId|gatewayServerId|uri|respCode|apiKeyId",
    },
    {
        "type": "top",
        "dimension": "apiId|appId|orgId|gatewayServerId|uri|respCode|apiKeyId",
        "limit": "3"
    },
    {
        "type": "pattern",
        "dimension": "uri",
        "patterns": [{
            "value": "uriPattern1",
            "name":"pattern_name"
        }]
    },
    {
        "type": "range",
        "dimension": "respCode",
        "ranges": [{
                "name":"",
                "from": "",
```

```
                    "to": ""
                },
                {
                    "name":"",
                    "from": "",
                    "to": ""
                }
            ]
        }
    ]
}
```

Response Fields

You can view the holistic picture when data is not time-bucketed in the Request input fields. When you use Time Range filters, you can view the response trends.

Parameter Name	Value
query	Specifies the response query.
data	Specifies the response data.

Example Response:

```
{
    "query": {},
    "data": [{
            "<dimension>": "<value>",
            "<aggregation>": "<value>",
            "buckets": [{
                "<aggregation>": "<value>",
                "date": "<DATE_1>"
            }, {
                "<aggregation>": "<value>",
                "date": "<DATE_2>"
            }]
        }
    ]
}
```

Examples

Total Hits for Default Time Range (Last 7 days)

Request:

```
{
    "metrics": {
        "type": "hits",
        "aggregation": "count"
    }
}
```

Response:

```
{
    "query": {
        "startDate": "2020-02-28T00:00:00",
        "endDate": "2020-03-05T10:34:47",
        "period": "7D(default)"
    },
    "data": [
        {
            "count": 373741.0
        }
    ]
}
```


Hits for Period (no aggregation) with Filter

Request:


```
{
    "metrics": {
        "type": "hits",
        "aggregation": "count"
    },
    "timeRange": {
        "type": "period",
        "period": "7D"
    },
    "filter": {
        "type": "and",
        "fields": [
            {
                "dimension": "gatewayServerId",
                "type": "in",
                "values": [
                    "40f30e9b-0e90-4736-8daf-684d7922b4a4"
                ]
            }
        ]
    }
}
```


Response:


```
{
    "query": {
        "startDate": "2020-02-28T00:00:00",
        "endDate": "2020-03-05T11:25:54",
        "period": "7D"
    },
    "data": [
        {
            "count": 186869.0
        }
    ]
}
```

Hits with Dynamic Time Range (Last 7 days) by Day

Request:

```
{
    "metrics": {
        "type": "hits",
        "aggregation": "count"
    },
    "timeRange": {
        "type": "period",
        "period": "7D",
        "aggregation": "day"
    }
}
```

Response:

```
{
    "query": {
        "startDate": "2020-02-28T00:00:00",
        "endDate": "2020-03-05T10:36:08",
        "period": "7D",
        "aggregation": "day"
    },
    "data": [{
        "count": 373741.0,
        "buckets": [{
            "date": "2020-03-04T00:00:00.000Z",
            "count": 49581.0
        }, {
            "date": "2020-03-03T00:00:00.000Z",
            "count": 65143.0
        }, {
            "date": "2020-03-02T00:00:00.000Z",
            "count": 64844.0
        }, {
            "date": "2020-03-01T00:00:00.000Z",
            "count": 64337.0
        }, {
            "date": "2020-02-29T00:00:00.000Z",
            "count": 64676.0
        }, {
            "date": "2020-02-28T00:00:00.000Z",
            "count": 65160.0
        }]
    }]
}
```

Top 3 APIs by API (filter errors) for a Dynamic Time Period (Last 7 days) Without any Time-grouping

Request:

```
{
  "filter": {
```

```
        "dimension": "respCode",
        "lower": 400,
        "lowerStrict": true,
        "type": "bound",
        "upper": 600,
        "upperStrict": false

  },
  "groupBy": [
    {
      "dimension": "apiId",
      "type": "default"
    },
    {
      "dimension": "appId",
      "limit": 3,
      "type": "top"
    }
  ],
  "metrics": {
    "aggregation": "count",
    "type": "hits"
  },
  "timeRange": {
    "period": "7D",
    "type": "period"
  }
}
```


Response:


```
{
  "query": {
    "startDate": "2020-02-28T00:00:00",
    "endDate": "2020-03-05T11:21:22",
    "period": "7D"
  },
  "data": [
    {
      "apiName": "Demo API 12a08972",
      "apiId": "faf3dd69-ed20-449c-abe7-84ae7a95bdef",
      "count": 1895
    },
    {
      "apiName": "Demo API 7a46de4a",
      "apiId": "dfc4df78-98ab-4253-b27d-b4b0b1be39f9",
      "count": 1843
    },
    {
      "apiName": "Demo API 53bcbe3e",
      "apiId": "9e4b1bd6-0e81-4da7-83b0-dd45c50d2e73",
      "count": 1807
    }
  ]
```

```
}
```

Hits with Static Time Range by Week with and, or, in, and bound Dimensional Filters

Request:

```
{
    "metrics": {
        "type": "hits",
        "aggregation": "count"
    },
    "timeRange": {
        "type": "interval",
        "interval": "2020-02-11T00:00:00/2020-03-05T23:59:59",
        "aggregation": "week"
    },
    "filter": {
        "type": "and",
        "fields": [
            {
                "dimension": "apiId",
                "type": "in",
                "values": [
                    "c9e29c47-4ac5-4464-b511-3aec4de99c89",
                    "3e80f77d-d99d-40fc-8bf8-7a8f4ee09aa8"
                ]
            },
            {
                "type": "or",
                "fields": [
                    {
                        "type": "bound",
                        "dimension": "respCode",
                        "lower": 200,
                        "upper": 299,
                        "lowerStrict": false,
                        "upperStrict": true
                    },
                    {
                        "type": "bound",
                        "dimension": "respCode",
                        "lower": 300,
                        "upper": 399,
                        "lowerStrict": false,
                        "upperStrict": true
                    }
                ]
            }
        ]
    }
}
```

Response:

```
{
    "query": {
        "startDate": "2020-02-11T00:00:00",
        "endDate": "2020-03-05T23:59:59",
        "aggregation": "week"
    },
    "data": [
        {
            "count": 230507.0,
            "buckets": [
                {
                    "date": "2020-03-02T00:00:00.000Z",
                    "count": 27943.0
                },
                {
                    "date": "2020-02-24T00:00:00.000Z",
                    "count": 70166.0
                },
                {
                    "date": "2020-02-17T00:00:00.000Z",
                    "count": 71165.0
                },
                {
                    "date": "2020-02-10T00:00:00.000Z",
                    "count": 61233.0
                }
            ]
        }
    ]
}
```


Hits with Dynamic Time Range with Dimensional Filters and groupBy (Default and URI Pattern)

Request:


```
{
    "metrics": {
        "type": "hits",
        "aggregation": "count"
    },
    "timeRange": {
        "type": "period",
        "period": "3D",
        "aggregation": "day"
    },
    "filter": {
        "dimension": "respCode",
        "type": "in",
        "values": [
            "200",
            "400"
        ]
    },
```

```
    "groupBy": [
        {
            "type": "default",
            "dimension": "respCode"
        },
        {
            "type": "pattern",
            "dimension": "uri",
            "patterns": [
                {
                    "type": "regex",
                    "value": "/accounts/v1/[d]/transactions[x]",
                    "name": "Transaction"
                },
                {
                    "type": "regex",
                    "value": "/account[x]",
                    "name": "Account"
                }
            ]
        }
    ]
}
```


Response:


```
{
    "query": {
        "startDate": "2020-03-03T00:00:00",
        "endDate": "2020-03-05T10:51:48",
        "period": "3D",
        "aggregation": "day"
    },
    "data": [
        {
            "uri": "Transaction",
            "respCode": "200",
            "count": 34375.0,
            "buckets": [
                {
                    "date": "2020-03-04T00:00:00.000Z",
                    "count": 15027.0
                },
                {
                    "date": "2020-03-03T00:00:00.000Z",
                    "count": 19348.0
                }
            ]
        },
        {
            "uri": "Account",
            "respCode": "200",
            "count": 17259.0,
            "buckets": [
```

```
                {
                    "date": "2020-03-04T00:00:00.000Z",
                    "count": 7321.0
                },
                {
                    "date": "2020-03-03T00:00:00.000Z",
                    "count": 9938.0
                }
            ]
        },
        {
            "uri": "Transaction",
            "respCode": "400",
            "count": 1920.0,
            "buckets": [
                {
                    "date": "2020-03-04T00:00:00.000Z",
                    "count": 842.0
                },
                {
                    "date": "2020-03-03T00:00:00.000Z",
                    "count": 1078.0
                }
            ]
        },
        {
            "uri": "Account",
            "respCode": "400",
            "count": 951.0,
            "buckets": [
                {
                    "date": "2020-03-04T00:00:00.000Z",
                    "count": 415.0
                },
                {
                    "date": "2020-03-03T00:00:00.000Z",
                    "count": 536.0
                }
            ]
        }
    ]
}
```


Latency for Dynamic Time Range (Last 3 Days) by Day with Dimensional Filters and Multiple groupBy (Top and Default)

Request:


```
{
    "metrics": {
        "type": "hits",
        "aggregation": "count"
    },
```

```
"timeRange": {
    "type": "period",
    "period": "3D",
    "aggregation": "day"
},
"filter": {
    "type": "and",
    "fields": [
        {
            "dimension": "appId",
            "type": "in",
            "values": [
                "acbc4c5c-ec1d-4196-88bc-ef09c3d95030"
            ]
        },
        {
            "dimension": "respCode",
            "type": "in",
            "values": [
                "200"
            ]
        },
        {
            "type": "or",
            "fields": [
                {
                    "type": "regex",
                    "dimension": "uri",
                    "pattern": {
                        "type": "regex",
                        "value": "/account[x]",
                        "name": "Account"
                    }
                }
            ]
        }
    ]
},
"groupBy": [
    {
        "dimension": "apiId",
        "type": "top",
        "limit": 2
    },
    {
        "type": "default",
        "dimension": "appId"
    },
    {
        "type": "default",
        "dimension": "respCode"
    }
]
```

```
}
```

Response:


```
{
    "query": {
        "startDate": "2020-03-03T00:00:00",
        "endDate": "2020-03-05T10:44:51",
        "period": "3D",
        "aggregation": "day"
    },
    "data": [
        {
            "apiName": "Demo API 612461",
            "apiId": "0df96035-3ee6-40e5-adfa-81f2efda86e6",
            "appName": "Demo Application 297293",
            "appId": "acbc4c5c-ec1d-4196-88bc-ef09c3d95030",
            "respCode": "200",
            "count": 425.0,
            "buckets": [
                {
                    "date": "2020-03-04T00:00:00.000Z",
                    "count": 170.0
                },
                {
                    "date": "2020-03-03T00:00:00.000Z",
                    "count": 255.0
                }
            ]
        },
        {
            "apiName": "Demo API 665797",
            "apiId": "9ea73d60-af28-449f-8531-040df19cab59",
            "appName": "Demo Application 297293",
            "appId": "acbc4c5c-ec1d-4196-88bc-ef09c3d95030",
            "respCode": "200",
            "count": 423.0,
            "buckets": [
                {
                    "date": "2020-03-04T00:00:00.000Z",
                    "count": 179.0
                },
                {
                    "date": "2020-03-03T00:00:00.000Z",
                    "count": 244.0
                }
            ]
        }
    ]
}
```


Latency for Dynamic Time Range (Last 365 Days) by Month Grouped by Response Code Ranges

Request:

```
{
  "groupBy": [
    {
      "dimension": "respCode",
      "ranges": [
        {
          "from": "200",
          "to": "299",
          "name": "Success Responses"
        },
        {
          "from": "400",
          "to": "499",
          "name": "Client Error Responses"
        },
        {
          "from": "500",
          "to": "599",
          "name": "Server Error Responses"
        }
      ],
      "type": "range"
    },
    {
      "dimension": "appId",
      "limit": 3,
      "type": "top"
    }
  ],
  "metrics": {
    "aggregation": "count",
    "type": "hits"
  },
  "timeRange": {
    "aggregation": "month",
    "period": "365D",
    "type": "period"
  }
}
```

Response:

```
{
  "query": {
    "startDate": "2019-03-07T00:00:00",
    "endDate": "2020-03-05T11:31:54",
    "period": "365D",
    "aggregation": "month"
  },
  "data": [
    {
      "appName": "",
      "appId": "",
      "respCode": "Success Responses",
```

```
    "count": 114211,
    "buckets": [
      {
        "date": "2020-03-01T00:00:00.000Z",
        "count": 12517
      },
      {
        "date": "2020-02-01T00:00:00.000Z",
        "count": 101694
      }
    ]
  },
  {
    "appName": "Demo Application d4d6d01c",
    "appId": "9417bc12-048b-4f84-980e-62e420a988da",
    "respCode": "Success Responses",
    "count": 56873,
    "buckets": [
      {
        "date": "2020-03-01T00:00:00.000Z",
        "count": 6294
      },
      {
        "date": "2020-02-01T00:00:00.000Z",
        "count": 50579
      }
    ]
  },
  {
    "appName": "Demo Application 65e3e69c",
    "appId": "36f618f5-cce3-40c1-8695-8712f3e26a99",
    "respCode": "Success Responses",
    "count": 56831,
    "buckets": [
      {
        "date": "2020-03-01T00:00:00.000Z",
        "count": 6402
      },
      {
        "date": "2020-02-01T00:00:00.000Z",
        "count": 50429
      }
    ]
  },
  {
    "appName": "",
    "appId": "",
    "respCode": "Client Error Responses",
    "count": 12671,
    "buckets": [
      {
        "date": "2020-03-01T00:00:00.000Z",
        "count": 1392
```

```
        },
        {
          "date": "2020-02-01T00:00:00.000Z",
          "count": 11279
        }
      ]
    },
    {
      "appName": "Demo Application d4d6d01c",
      "appId": "9417bc12-048b-4f84-980e-62e420a988da",
      "respCode": "Client Error Responses",
      "count": 6316,
      "buckets": [
        {
          "date": "2020-03-01T00:00:00.000Z",
          "count": 689
        },
        {
          "date": "2020-02-01T00:00:00.000Z",
          "count": 5627
        }
      ]
    },
    {
      "appName": "Demo Application 65e3e69c",
      "appId": "36f618f5-cce3-40c1-8695-8712f3e26a99",
      "respCode": "Client Error Responses",
      "count": 6308,
      "buckets": [
        {
          "date": "2020-03-01T00:00:00.000Z",
          "count": 686
        },
        {
          "date": "2020-02-01T00:00:00.000Z",
          "count": 5622
        }
      ]
    }
  ]
}
```


Login API

The Login API provides programmatic access to the Portal API (PAPI), allowing you to integrate login and authentication flows directly in your custom application. To access the API, request an access token or revoke an access token by accessing the OAuth 2.0 authorization endpoint.

SSO (SAML or CA SSO) users cannot log in to API Management SaaS using the Login API; they have to use the API Management SaaS login page.

You cannot use PAPI to edit users who are mapped to multiple organizations.

You can retrieve the following using the Login API:

- The scope and identifier of an application that was created through the Mobile Developer Console (MDC) by accessing the `/application/{tenantId}` endpoint
- The session status of an OpenID application.
- The details of an OpenID application user.

All requests and responses are in JSON format.

Get an Access Token

Follow these steps:

- Retrieve the client_id (`apikey`) from the `/application/<tenantId>` endpoint.
- Use the retrieved client_id and use the `/login/auth/oauth/v2/authorize` endpoint.

Retrieve client_id

The following example accesses the `application` RESTful resource from the `apim-ssg.dev.ca.com` CA API Gateway:

```
curl -k https://apim-ssg.dev.ca.com:8443/login/application/apim
```

A response similar to the following example is expected:

```
{
    "apikey": "1234abc1c20ea555b59def43f7ebf01234",
    "scope": "openid"
}
```

Use the client_id

Use the client_id with the OAuth 2 resource password flow, but do not include the client_secret.

The following code is an example of how to use the API:

```
curl 'https://apim-ssg-apim-trial1-uswest2.app.services.ostest1.dev.ca.com/login/auth/
oauth/v2/token' --data
'client_id=5a6050112e77410a9ce2276c7c709643&grant_type=password&scope=OOB&username=sell2\test&p
&login_hint=sell2'
```

Get Session and User Information

You can use the access token to retrieve session information and user info for a specific user.

Follow these steps:

1. Retrieve session and user information for a specific user from the `/openid/connect/v1/userinfo` endpoint.
2. Validate session information from the `/connect/session/status` endpoint.

Explore the Login API

Follow these steps:

1. Go to Portal and select **Publish**.
2. Select **API Explorer**.

   NOTE
   API Explorer is only accessible through the API Portal/Ingress tenant.

3. Select **Login API** from the **API** drop-down list.
   Alternatively, download the login-swagger-4-1.json file, and then open the file in a Swagger UI console or editor.

Authorization API

The Authorization API provides programmatic access to API Portal entities which let you perform operations related to role-based access control. For example, you can add permissions for a role to delete applications. This will allow users who have that role in a given organization to be able to delete applications. You can call the Authorization API from your external client application, or you can try it out using the API Explorer.

> **TIP**
>
> **Have you considered Org Publisher?**
>
> In API Portal release 4.5 and higher, the new Org Publisher role allows you to grant CRUD permissions to users belonging to a specific organization. You might want to use this new role in lieu of the Authorization API. For more information about the Org Publisher role, see User Types, Roles and Permissions.
>
> **NOTE**
>
> You can add or remove the following permissions to the organization administrator role: Create, update, and delete APIs in their organization using the Authorization API. The create, update, and delete privileges are not available by default for this role. You cannot remove default permissions from the role using the Authorization API.

The following RESTful resources are included in the API:

- **Permitted**
  Use this resource to check if a permission (such as the ability to delete APIs) is granted for a specific role (such as API Owner).
- **Roles**
  Use this resource to retrieve all roles, retrieve all permissions for a specific role, and retrieve or update permissions to a specific entity (such as API).

Requests and responses are in JSON format.

You can use the Authorization API to distribute role memberships in your organization. Role permissions are reflected in the Portal interface, as well as while making Portal API (PAPI).

Access the Authorization API using the API Explorer.

You can try out the Authorization API using the API Explorer.

Follow these steps:

1. Log in to the API Portal as a Portal administrator for the intended tenant.
2. Select **Portal API**.
3. Select the **Portal Authorization API** option from the **API** drop-down list.

The API Explorer appears in the right pane, showing the Authorization API. You have access the PAPI using the API Explorer.

Authentication

Authorization API calls require a valid OAuth token.

For more information about OAuth tokens, see the OAuth Community site.

Example

You can download a sample Swagger JSON file describing the current Authorization API from ../../../../../../assets/docops/apiportal/rbac-swagger-4-1-4.json.

For more information about the Swagger (OpenAPI) specification, see the Swagger website.

API Hub

NOTE

API Hub for Portal Versions 5.1.2 or Newer: Inline JavaScript Prohibited

API Hub is a react-admin-based implementation for the developer console of Layer7 API Developer Portal (API Management SaaS). The standard API Hub is a reference implementation and is included with API Management SaaS.

Watch this video to learn about API Hub:

You can use the standard API Hub out of the box, or you can have your UI development team provide a customized API consumer-facing user experience of API Management SaaS by customizing and extending the standard API Hub. Use the standard API Hub when you have minimal customization, marketing, and branding requirements. This is the ideal option for internal-facing API Management SaaS, where you have minimal marketing and branding requirements for your developer console of API Hub. Customize API Hub if you require localization, custom pages, branding, themes, and images/logos. The Portal Admin determines the requirements for the API consumer experience when using API Management SaaS.

NOTE
Only Portal Admins can change the email address after authenticating with the login password in the My Profile page. All the other users such as Idp users and developers cannot update their own email address.

This section includes the following articles:

Determine the API Hub to Use

Use the following table to guide your decision of when to use the standard API Hub or when to customize it:

Developer Console Requirement	Standard API Hub	Customized API Hub
Minimal customization and branding	✓	✗
Localization/language support	✓	✓
Branding with customized logos	✗	✓
Custom pages	✗	✓
Custom navigation and flow	✗	✓
Custom themes per organization	✗	✓

Supported Languages

API Hub supports English, Spanish, and French languages.

Accessibility

Broadcom is committed to ensuring that all customers, regardless of ability, can successfully use its products and supporting documentation to accomplish vital business tasks.

Getting Started with API Hub

This topic is intended to serve as a guide and help new API Hub users quickly get started on understanding the requirements and best practices in setting up the API Hub

Contents:

- Prerequisites
- Setup Your Local Environment
- Customize
- Build and Deploy
- Localizing Content
- User Logins
- Troubleshooting
- Best Practices

Prerequisites

You can download the API Hub reference implementation in GitHub here.

Required Skills Set

As this is a Front-end application, you will need mostly Front-end development knowledge such as:

- React/React-admin/Redux
- Javascript/html/css
- Node/yarn
- npm package manager
- Cypress/jest/react-testing library

Required Components and Repository Overview

This project is using the react-admin library which is based on React. The folder structure is divided into 4 main packages:

- Packages > Example App - The Example app demonstrates the usage of the Layer7 API Hub library and is the reference implementation for API Hub. It is used for Layer-7-mock data.
- Packages > Healthcare: The Healthcare app further extends the Example app with more customizations and a Healthcare theme. It includes the same developer features as the Example app plus custom pages and additional API calls (PAPI and Portal Metrics API).
- Packages > Layer7-apihub: The Layer7 API Hub Library contains all the React components used by API Hub. It includes components that handle API and Application management, authentication, documentation, user management, and more.
- Packages > layer7-apihub-mock: The purpose of the Layer7 API Hub mock server is to mock PAPI responses that are returned back to the API Hub frontend. It runs in a web browser and intercepts calls from the frontend and returns predefined responses. The Healthcare app is used with this mock server.

Most of the folders name are explicit in their purpose., For example, in the example project, the folders are described as follows:

- **config**: used for configuration
- **public**: contains publicly available files such index.html, logo or favicon
- **node_modules**: contains all dependencies
- **src**: location where most of the code is run This is where the logic resides and most of the work and customization will happen.

Setup the Local Environment

To run the project locally on Mac OS, follow the steps below:

1. Clone the APIHub repository:
   ```
   $ git clone https://github.com/CAAPIM/APIHub.git
   ```
2. Install Node Version Manager (NVM) on your machine to manage the Node JS version you want to run.
3. In the command line, navigate into the project `/APIHub` and run the following command to start using the correct node version:
   ```
   $ nvm use
   ```
   If the version of node isn't installed yet on your machine, follow the suggested command prompts to install that specific version.
4. Install yarn globally:
   ```
   $ npm install --global yarn
   ```
5. Install the dependencies of the project:
   ```
   $ make install
   ```
6. To start the example app, run the following:
   ```
   $ make start
   ```

Customize

After successfully creating your local environment for API Hub, you can now begin your own customizations and development of API Hub. To learn how you can add pages, apply new themes, and host your custom API Hub, see Customize and Extend the Standard API Hub.

Build and Deploy

After configuring and customizing your API Hub, you'll want to build, deploy, and test the project. This front-end project can be built statically and then deployed. Here are a few steps to help you understand how to build and deploy it.

Build

Build the API Hub library, the Example app, and then copy the production configuration by running the following commands:

```
$ make build
$ make build-example
$ DEPLOY_ENV=prod make copy-deploy-config-example
```

Deploy

After building the project, copy the contents that have been generated in the `packages/example/build` directory to your favorite web hosting service.

For example, the following command launches an NGINX Docker container on your local machine:

```
$ docker run --name APIHub -v `pwd`/packages/example/build:/usr/share/nginx/html:ro -p 8888:80 nginx
```

Test

The end-to-end (E2E) tests are designed for the Example app. For more information about testing, see the Cypress -- End-To-End Testing README.md in the Cypress directory.

Localizing Content

API Hub supports English, Spanish, and French languages out of the box. Existing translation files are located in `packages/layer7-apihub/src/i18n` .

To implement a new language for localization of the API Hub (e.g., Italian) follow the steps below:

1. Using an existing language .js file as a template, create the .js file with the appropriate translations. For example,for Italian, we've created a new it.js file in the i18n folder: `packages/layer7-apihub/src/i18n/it.js`
2. Add the new language line like the other languages in: `packages/layer7-apihub/src/i18n/index.js` For our new Italian entry, we added the following lines:
```
import it from './it';
…
export const italianMessages = it;
```
3. Next, we'll import the italianMessages file created from the previous step and add an IF statement in i18nProvider. Go to: `packages/example/src/i18n/i18nProvider.js`
4. Add the following IF statement (follow other languages IF statements as examples in file):
```
locale => {
    if (locale === 'it') {
        return italianMessages;
    }
```
5. Add the new language as part of the supported locales in: `packages/layer7-apihub/src/i18n/supportedLocales.js`
   For example:
```
export const supportedLocales = {
    en: 'English',
    it: 'Italian',
};

export const documentationLocales = {
    en: 'en-US',
    it: 'it-IT',
};
```
6. Following through with our Italian language support example, we'll create another it.js file in the example package folder:
   a. Go to `packages/example/src/i18n/it.js` .
   b. Use an existing language file as a template (e.g., en.js) and replace the values to import italianMessages from layer7-apihub.
   c. Optional: You can add additional translations in this file that will merge with translations from layer7-apihub .
7. Import the supportedLocales.js file as modified in step 5 and add an IF statement in i18nProvider as described in that step (e.g., to return italianMessges), located in: `packages/example/src/i18n/i18nProvider.js`

After completing the steps above, your custom translation shall be available as a selectable option in the API Hub.

User Logins

The current implementation of the API Hub uses mock data to render the example. You may connect it to your own server.

You are free to login using one of the following methods: Username/Password, LDAP, or SAML. Alternatively, you can create your own account by clicking the Create an API Hub Account button on the bottom left.

As for the different roles and users, here are the presets that have been implemented and are extendable with their corresponding test login credentials:

Account	Login	Password
Portal Admin	portalAdmin	Password@1
API Owner	apiOwner	Password@1
Org Publisher	orgPublisher	Password@1
Org Admin	Publisher	Password@1
Developer	user	Password@1

Troubleshooting Resources

Since API Hub is a React application, you may reference React debugging best practices as outlined in React Design Principles.

An browser extension called React DevTools is a browser extension that can help you debug your React apps such as API Hub. For more information, see this React blog post.

At a more fundamental level, debugging with the console.log() method may be sufficient for testing and diagnosing errors found in any React application.

Best Practices

The following are resources for best practices that can be referenced when developing in React:

Thinking in React	React is component oriented and it's recommended that you check out their Thinking in React documentation on how to think about building components and applications in a multi-step process.
Linters	Linter is a static code analysis tool used to catch programming errors and can used for the development and customization of React applications. There are several flavors of this tool, such as Eslint.
File Naming Convention	It's widely accepted that properly and consistently-named file or folder can help yourself and others understand your code more quickly. For tips on naming conventions in React, see this Upbeat Code article.
File Structure	React has several recommendations on how you should structure your React projects - see File Structure in their documentation to learn more.

The following are some general development best practices and recommendations for API Hub development in general.

Reuse Existing API Hub Components	Reuse of components in a React project aligns with the 'Thinking in React' philosophy. Components you'll reuse the most for your API Hub project reside in `packages/layer7-apihub` .
Use the React Admin Component	The <Admin> component creates an application with its own state, routing, and controller logic and can be used for your API Hub project as a component. See The Admin Component to learn more.
Use the Material UI Component	The current API Hub reference implementation uses Material UI as a dependency to create components that make up the user interface. Instead of using multiple UI libraries, it's generally a good idea to commit to one third-party library, sch as Material UI, since they typically provide ready-to-use components.
Use Themes	API Hub uses themes to customize the front-end. See Customize and Extend the Standard API Hub to learn more.

Configure the Standard API Hub at Runtime

You can perform the following basic configurations and branding of the standard API Hub at runtime:

- Manage the content that is displayed on the Home page of the developer console. This can include branding and marketing content.

  NOTE
  Limit changes on the Home page to text and headings. Do not create horizontal rules and tables in the markdown. If you have additional layout requirements for the Home page, customize and extend the standard API Hub.
  For more information, see Customize and Extend the Standard API Hub.
- Manage localized API documents.
- Manage wiki documents.
  For more information about how to manage wiki documents, see Manage the Wiki Documents in API Hub.
- Manage the overview content for applications.

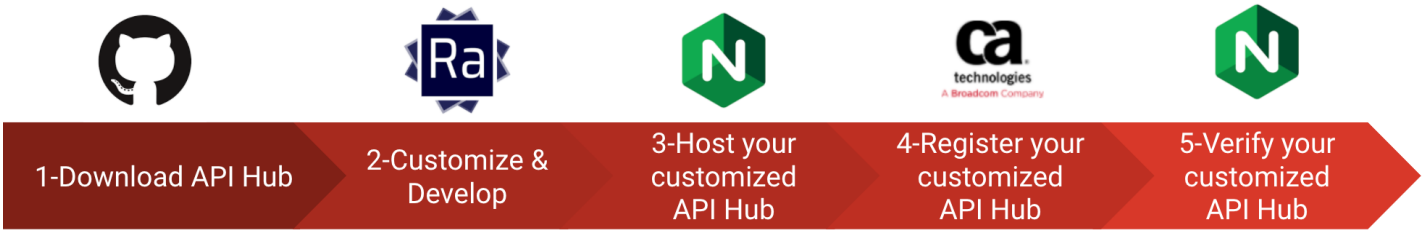
For more information about how to perform these configurations, see Manage the Content in API Hub.

Customize and Extend the Standard API Hub

Your API Management SaaS team can customize API Hub by designing, creating, and hosting a customized version of API Hub. Enhance (develop and test) as a normal UI development project.

The standard API Hub returns API Management SaaS data by referencing the Portal API (PAPI). You can customize and extend the standard API Hub by adding custom code. You can further enrich your developer experience by including calls to the PAPI in the custom code. Ensure that this custom code renders properly.

The following image shows the workflow for customizing API Hub:



In this article:

Download API Hub

Prerequisite: You have read the API Hub README.

Follow these steps:

1. (Optional) Fork the public read-only repo of API Hub from the API Hub GitHub.
2. Clone the repository to your local development environment.

You have downloaded the Example app from GitHub. This app is a reference implementation of API Hub, including the components that are packaged into API Hub. It has the same source code as the standard API Hub.

API Hub Apps and Packages

Example app

The Example app demonstrates the usage of the Layer7 API Hub library and is the reference implementation for API Hub. It has the same source code as the standard API Hub that comes with Layer7 API Developer Portal and is built on top of Create React App (CRA).

Healthcare app

The Healthcare app further extends the Example app with more customizations and a Healthcare theme. It includes the same developer features as the Example app plus custom pages and additional API calls (PAPI and Portal Metrics API).

Layer7-apihub-mock

The purpose of the Layer7 API Hub mock server is to mock PAPI responses that are returned back to the API Hub frontend. It runs in a web browser and intercepts calls from the frontend and returns predefined responses. The Healthcare app is used with this mock server.

Layer7-apihub

The Layer7 API Hub Library contains all the React components used by API Hub. It includes components that handle API and Application management, authentication, documentation, user management, and more.

Customize and Develop API Hub

Start your own customizations and development of API Hub. You can check in and maintain your code in your own repository. To make updates easier, we recommend that you start your own package.

Add Pages

You can add pages in the Layer7 API Hub library or in an application using the Layer7 API Hub library. The add pages mechanism is based on react-admin custom routes and the React Router library. Follow the sample code to add a new page to your API Hub.

Theme Customizations

You can apply new themes to your API Hub by modifying the default themes in the Example app. By default there are two themes available, and you can toggle between them in the UI. See theme.js for the implementation.

For more information about the customizations that you can make in API Hub, see the API Hub README.

Develop and Test Your Customizations

Develop and test as a normal UI development project. Verify your customizations using the API endpoint responses that the API Hub mock server provides. The mock server improves the API Hub developer experience by sending API responses that mimic PAPI without having to connect to a full API Management SaaS stack.

For more information about the mock server, see the API Hub Mock Server Package README.

Host your Customized API Hub

Host your customized API Hub using your own hosting solution. Ensure that you have properly defined your hosting environment in your configuration. Work with your Portal Admin to confirm the parameter values.

Prerequisite: You have a static web hosting solution.

Configurations for your customized API Hub are defined in the `config.js` file. For more information on defining this file for a customized API Hub, see the API Hub Example App GitHub page.

Register the Hosting Domain of your Customized API Hub with API Portal

Register the hosting domain of your customized API Hub with API Management SaaS so that it can successfully communicate with API Management SaaS and so that you can alleviate CORS-restriction issues. Do this step for each customized API Hub.

> **NOTE**
> The Ingress cache can take some time to update. If you are unable to view your customized API Hub on the hosting domain, wait a few minutes for the update, and then refresh.

Issue a PUT request to the following PAPI `Settings` endpoint, with the following payload:

```
https://${tenant_id}.${hostname}/api/${tenant_id}/Settings('APIHUB_SETTINGS')
```

Where:

- `${tenant_id}.${hostname}` is the tenant URL.
- `${tenant_id}` is the tenant for which API Hub is configured.

For example:

```
https://apim.dev.ca.com/api/apim/Settings('APIHUB_SETTINGS')
```

Payload Example #1:

The following payload example registers the hosting domain for the `apihub1` customized API Hub. The name of the customized API Hub displays on, and is accessible from, the menu bar

```
{
"Name": "APIHUB_SETTINGS",
"Value": "[{\"name\": \"apihub1\",\"host\": \"developer1.example.com\",\"forgetpasswordPath\": \"/#/new-
password\",\"signuppath\": \"/#/account-setup\"}]",
"Uuid": "<GENERATED_UUID>"
}
```

Payload Example #2:

The following payload example registers the hosting domain for the `apihub1` and `apihub2` customized API Hubs:

```
{
"Name": "APIHUB_SETTINGS",
"Value": "[{\"name\": \"apihub1\",\"host\": \"developer1.example.com\",\"forgetpasswordPath\": \"/#/new-
password\",\"signuppath\": \"/#/account-setup\"},{\"name\": \"apihub2\",\"host\":\"developer2.example.com\",
\"forgetpasswordPath\": \"/#/new-password\",\"signuppath\": \"/#/account-setup\"}]",
"Uuid": "<GENERATED_UUID>"
}
```

> **IMPORTANT**
> - The URL for the host must not contain a "/" at the end of the URL.
> - The "Value" attribute must be one single line.

> **NOTE**
> You can register more customized API Hubs by appending to the list of already registered API Hubs in the payload, and then issuing a PUT request to the endpoint. Avoid overwriting the already registered API Hubs' data and losing connectivity by only appending to the list of API Hubs.

The “Value” attribute maps to a JSON array where each element contains the settings for one API Hub. The settings are as follows:

- `name` matches the `ORIGIN_HUB_NAME` parameter that you defined in the `config.js file` .
- `host` is the fully qualified domain name where your customized API Hub is hosted.
- `forgetpasswordPath` is the path for the Forgot Password page for your customized API Hub. This parameter is required but you can leave the value empty. For example, in the previous example, the Forgot Password page for the customized API Hub uses the following URL: `https://developer2.example.com/apihub1/#/new-password`
- `signuppath` is the path for the new user Sign Up page. This parameter is required (even if third-party registration is disabled). You can leave the value empty.

Verify your Customized API Hub

Access your customized API Hub and verify that it works as intended.

For more information about how to access your customized API Hub, see Access API Hub.

Next Steps

Now that you have customized and extended the standard API Hub, you can share your customized API Hub site with your API consumers.

Access API Hub

The method you use to access API Hub depends on your role. Publishers (Portal Admins, API Owners, and Org Publishers) typically access API Hub from API Management SaaS. Org Users (Org Publishers, Org Admins, and Developers) and other API consumers, such as partners, access API Hub directly using a URL.

Access API Hub from API Portal

You can access the standard API Hub from API Management SaaS.

Follow these steps:

1. Log in to API Management SaaS as a Portal Admin.
2. Do one of the following:
   - (If only the standard API Hub exists) To access the standard API Hub, from the menu bar, select **API Hub**.
   - (If your API Management SaaS team has customized and extended API Hub):
     - To access a customized API Hub, from the menu bar, select **API Hub**, and then select the customized API Hub that you want to access from the drop-down.
     - To access the standard API Hub, select **API Hub (default)** from the drop-down.

Access API Hub Directly

You can access the *standard* API Hub by appending `/apihub/#/login` to the end of the API Management SaaS URL. For a *customized* API Hub, you can access it using the URL for the customized API Hub site that the API Management SaaS team shared with you. The URL includes the registered hostname, for example:

```
http://developer1.example.com/apihub/#/login
```

.

If you are unsure of the URL for the customized API Hub site, contact the Portal Admin.

Okta Single Sign-on for API Hub

API Hub is integrated with Okta single sign-on capability.

Click **SSO LOGIN** on the API Hub sign-in page. Enter your email address to access the Okta widget that authenticates the identity provider (IdP). After successful log in using your Okta credentials, the API Hub site opens.

Manage the Content in API Hub

As a Portal Admin or API Owner, you can manage the content in API Hub.

In this article:

Switch API Hub to Another Language

By default, when you first log in to API Hub, the language is English. You can change the language for API Hub to one of the languages that API Hub supports using the language selector at the top of the page:

API Hub remembers the language that you last selected for API Hub, so that the next time that you log in to API Hub, that language is already selected.

Manage the Content on the Home Page

You can manage (add and edit) the content that is displayed on the Home page of the developer console. By default, API Hub displays Home page content based on the language that you have selected for API Hub. You can add content to the Home page in each of the languages that API Hub supports.

The content that you add to the Home page is in the language that you have selected for API Hub. To add content in another language that API Hub supports, switch to that language using the language selector at the top of the page.

Follow these steps:

1. Sign in to API Hub as a Portal Admin.
   The Home page appears.
2. Click Edit (the pencil icon).
3. Add markdown content for the Home page in the pane to the left, the edit pane, and then save your changes.
   The edit pane is a markdown editor. The edit pane and the pane to the right, the preview pane, are a side-by-side view of the overview. You can:
   – Copy and paste markdown into this pane.
   – Format the content that you add using the options on the menu bar or using markdown basic syntax.
   For more information about markdown basic syntax, see the Markdown Basic Syntax Guide.

Manage Localized API Documents

You can manage (add, edit, and delete) English API documents using API Management SaaS and using API Hub. However, if you have localization requirements, manage the documents in the other languages that API Hub supports using API Hub.

> **TIP**
> You can also manage API documents by making calls to the PAPI `Documents` resource.
> For more information about this resource, see Portal API (PAPI).

For more information:

- About how to manage API documents using API Management SaaS, see Manage API Documents.
- About the languages that API Hub supports, see API Hub.

Select the Language for API Documents

The English-language API documents that are published in API Management SaaS also display in English in API Hub. You can manage English-language documents using both API Management SaaS and API Hub. If you have localization requirements, manage these localized documents using API Hub. This allows you to view the API Hub developer experience cohesively based on the selected language.

The language selector that is in the Documents tab is independent of the language selector for API Hub. By default, the selected language in the language selector that is in the Documents tab is the same as the selected language for API Hub. For example, if you are using API Hub with English as the selected language, and you want to add a Spanish-language document, switch to Spanish using the language selector that is in the Documents tab, and then add the Spanish-language document.

Navigate the Document Tree

In addition to using the mouse, you can navigate the document tree using the tab and arrow keys on your keyboard. Move the selector to the document tree using the tab key, then use the arrow keys to navigate through the document tree.

Reorder a Document

You can reorder documents within the document tree, such as moving a document to be a child document of another document or reordering the list of documents.

To reorder using the mouse, from the list of documents, click and hold the document that you want to reorder, and then drag and drop it to the new location in the tree.

To reorder using the keyboard, from the list of documents, pick up the document that you want to reorder by pressing the Spacebar key on your keyboard, navigate to the new position using the arrow keys on your keyboard, and then drop it to the new location in the tree by pressing the Spacebar key again.

View APIs using API Hub

You can view APIs and API documents using API Hub.

In this article:

View an API using API Hub

You can view the list of APIs to which you have visibility permissions using API Hub.

Follow these steps:

1. Sign in to API Hub.
   The Home page appears.
2. Select **APIs**.
   A list of APIs displays on the APIs page.
3. Select the tile for the API that you want to view.
   The API Details page appears.

View an API Document using API Hub

You can view the list of documents for those APIs to which you have visibility permissions using API Hub.

Follow these steps:

1. From the APIs page, click the API for which you want to view API documents.
   The Overview information opens.
2. Select the **Documentation** tab.
   A list of API documents displays. Those that are displayed are those that correspond to the language that you have selected for API Hub. For example, if you have selected Spanish as the language for API Hub (from the language selector at the top of the page), but the API only includes English-language API documents, an empty list displays. To view the documents in the other languages that API Hub supports, switch API Hub to that language.

   > **NOTE**
   > Your English-language documents are not automatically translated and published into the language that you have selected for API Hub.

View the API Swagger Documentation using API Hub

You can view the API Swagger documentation and test the API using API Hub.

Follow these steps:

1. From the APIs page, click the API for which you want to view API Swagger documentation.
   The Overview information opens.
2. Select the **Specs** tab.
   The Swagger UI opens.
3. From the **Search** or **Select Application** text box, enter the first few characters of the application's name for which you want to test and explore the API.
   A drop-down appears with the results that match closest to your search.
4. Select your API key.
   The API key (client ID) and the shared secret (client secret) (if the shared secret was generated in plaintext), display.
5. Ensure that your session is authorized. If applicable, the Padlock icon next to your selected endpoint indicates whether an endpoint is locked. If required, authorize your session by clicking the Padlock, and then completing the information required in the **Authorization** window.
6. Expand the endpoint that you want to execute, and then click **Try it out**.
   The example values in the **Request Body** field become editable.
7. Make changes to the example request, and then click **Execute**.

Manage Applications using API Hub

Applications are containers of related APIs in API Management SaaS. Org Users (Org Publishers, Org Admins, and Developers) can access those APIs while building your web/mobile application using the application. Org Publishers and Org Admins can manage the applications using API Hub by adding, or registering, them, by editing them, by controlling which applications use specific APIs, by generating new secrets, and by deleting them. Manage applications, for example, when testing your APIs and API Portal.

In this article:

> **NOTE**
> Except for viewing applications, the following workflows are available for Portal Admin, API Owner, Org Admin, and Org Publisher. Note that global publishers (Portal Admin and API Owners) can also continue to create and manage applications in API Portal.

View an Application using API Hub

All users can view applications using API Hub. Publishers can view applications for all organizations using API Hub. Org users, including developers, can view applications for their specific organization through API Hub.

Follow these steps:

1. From the list of applications, select the tile for the application that you want to view.
   The Application Details page appears in view-only mode.
2. From this page, you can do the following:
   - View the overview of the application. The application overview is additional information about the application.

     TIP
     For more information about the overview and how to manage this content, see the "Manage the Overview Content for an Application" section.
   - View the details of the application, including the organization and description.
   - View the APIs and API groups/API plans that have been added to the application.
   - View the authentication and credential information for all the API keys created under the application. If multiple keys are present, the default key is identified.

Manage the Overview Content for an Application using API Hub

For all languages that API Hub supports, the Portal Admin can manage the overview content for applications by adding, editing, and deleting the content.

Follow these steps:

1. Sign in to API Hub as Portal Admin.
   The Home page appears.
2. From the Home page, select **Applications**.
   The list of applications appears on the Applications page.
3. Select the tile for the application for which you want to add, edit, or delete an overview.
   The Application Details page appears in view-only mode.
4. The overview content that you add or edit is in the language that you have selected for API Hub. To add or edit overview content in another language that API Hub supports, switch to that language using the language selector at the top of the page.
5. In the **Overview** section, click **Edit overview** (the pencil icon).
   The **Overview** page appears.
6. Add or edit the markdown content for the overview of the application in the pane to the left, the edit pane, and then save your changes.
   The edit pane is a markdown editor. The edit pane and the pane to the right, the preview pane, are a side-by-side view of the overview. You can:
   - Copy and paste markdown into this pane.
   - Format the content that you add using the options on the menu bar or using markdown basic syntax.

   For more information about markdown basic syntax, see the Markdown Basic Syntax Guide.

The overview content for the application is saved.

Add an Application using API Hub

The applications that you add using API Hub are also available in API Management SaaS.

Follow these steps:

1. Sign in to API Hub as Portal Admin, API Owner, Org Admin, or Org Publisher.
   The Home page appears.

2. From the Home page, select **Applications**.
   The list of applications appears on the Applications page.
3. Select **Add Application**.
   The Add Application page appears.
4. Provide details about the application. Provide a unique application name and optionally a description.
5. In the **API Management** section, add or remove available APIs and API groups/API plans to or from your application.
   In addition to the listed APIs and groups/plans, you can search using the search field.

   NOTE
   If API plans are turned on, then API groups do not display.

   Do the following:
   – **To remove a selected API or API group/API plan from the application**, click the trash can icon for the API or API group/API plan that you want to remove. The list of selected APIs and API groups is under the **Selected APIs** and **API Groups** section.
   – **To add an available API or API group/API plan to your application**, click the plus icon to the left of the API or API group/API plan that you want to add, and then accept the terms and conditions of the end-user license agreement (EULA). The list of available APIs and API groups/API plans is under the **Available APIs** (or **Available API Groups/Available API Plans**) section.
     When you add an API group/API plan to your application, you add the APIs that are contained within the group/plan to your application. These APIs are enabled and public. If the APIs that are contained within the group/plan are enabled but private, then the APIs belong to your organization and have been added to the account plan that your organization uses.
     Prerequisite: You must have explicit access to the API or the API must belong to your organization.
6. If any of the APIs that you have added to the application use OAuth, in the **Authentication and Credentials** section, complete the following fields:
   – **Callback/Redirect URL(s)**
     Defines the callback/redirect URLs for your application. Separate multiple URLs using a comma.
     ```
     https://{yourportalurl}/admin/oauthCallback
     ```
   – **Scope**
     Defines the OAuth scope parameters that specify the privileges that this application requires from the protected APIs. Separate parameters using a space.
   – **Type**
     Defines the grant type for the OAuth-protected APIs that the application consumes.
     Values:
     - **None**
     - **Public:** Defines that the OAuth-protected APIs that this application consumes use the Implicit grant type.
     - **Confidential:** Defines that the OAuth-protected APIs that this application consumes use the Confidential grant type.

     Default: None
   – **Shared Secret Format**
     Determine the format for the shared secret for this application.
     Values:
     - **Plain text secret:** Generate the secret in plaintext format.
     - **Hashed secret:** Generate the secret in hashed format.

     Depending on API Management SaaS settings, Plain text Secret and/or Hashed Secret formats might be available. For more information about hashed secrets, see Enable Hashed Client Secret.
7. Select **Save**.

The application is added.

Edit an Application using API Hub

Follow these steps:

1. Sign in to API Hub as Portal Admin, API Owner, Org Admin, or Org Publisher.
2. From the list of applications, for the application that you want to edit, click the ellipsis in the tile, and then select **Edit**.
   The Application Details page appears in edit mode.

   NOTE
   You can also access this page by clicking **Edit** (the pencil icon) while viewing the page in view-only mode.
3. Edit the application, and then select **Save**.

Your changes are saved.

Generate a New Secret for an Application using API Hub

For those applications to which you have access, you can generate a new secret for your application, for example, if the shared secret is compromised.

WARNING
Generating a new secret changes the API key and voids the current API key. This breaks access for anyone using the current API key. Share and use newly-generated secrets with developers coding their application that uses the APIs.

Follow these steps:

1. While logged in to API Hub as Portal Admin, API Owner, Org Admin, or Org Publisher, from the Application Details page, click **Generate New Secret**, and then click **Plain text secret** or **Hashed secret**.
   For more information about hashed client secrets, see Enable Hashed Client Secret.
   The Generate New Secret window opens.
2. Click **Generate New Secret**.
3. Save your changes.

Delete an Application using API Hub

Follow these steps:

1. Sign in to API Hub as Portal Admin, API Owner, Org Admin, or Org Publisher.
2. From the list of applications page, for the application that you want to delete, click the ellipsis in the tile, and then select **Delete**.
3. Confirm the deletion by selecting **Delete**.

The application is deleted.

Manage Wiki Documents in API Hub

As a Portal Admin, you can publish markdown content as generic documents to your tenant in API Hub for all users to consume. You can add wiki documents in the languages that API Hub supports.

TIP
The following procedures describe how to manage wiki documents by way of API Hub. You can also manage wiki documents by making calls to the Portal API (PAPI) `Documents` resource, setting the `type` attribute to 'custom' and the `typeUuid` attribute to 'wiki1'.
For more information about this resource, see Portal API (PAPI).

In this article:

Add a Wiki Document

Follow these steps:

1. Sign in to API Hub, and then click **Wiki**.
   If API Hub includes wiki documents, a list of them appears. Otherwise, the list is empty.
2. The wiki documents language selector is independent of the language selector for API Hub (at the top of the page). By default, the selected language in the wiki documents language selector is the same as the selected language for API Hub. For example, if you are using API Hub with English as the selected language, and you want to add a Spanish-language wiki document, switch to Spanish using the wiki documents language selector, and then add the Spanish-language wiki document.
3. Complete one of the following:
   – To add a top-level document, click the Add Document icon (the + icon).
   – To add a child document nested beneath a parent document, click the ellipsis icon to the right of the parent document, and then select **Add Document**.
4. Add markdown content to your wiki document in the pane to the left, the edit pane.
   The edit pane is a markdown editor. The edit pane and the pane to the right, the preview pane, are a side-by-side view of the document. You can:
   – Copy and paste markdown into your document.
   – Format the content that you add using the options on the menu bar or using markdown syntax.
     For more information about markdown syntax, see the Markdown Guide.
5. Click **Add Document**.
   The Publish Document window opens.
6. Enter the following metadata for the document, and then click **Save**:
   Title
   The title for the document. This is what shows on the document tree.
   Allowable characters: Alphanumeric characters, dashes, underscores, and spaces
   URI
   The Uniform Resource Identifier (URI) is part of the URL. It identifies this document. By default, the URI includes the encoding that you can remove.
   Allowable characters: Alphanumeric characters, dashes, underscores, and spaces

   > **NOTE**
   > You cannot change the URI once you save this document. Spaces are replaced with underscores.

The wiki document is added to the list of documents.

> **NOTE**
> You can view the URI for this newly-added document in the web browser's address bar.

Edit a Wiki Document

Follow these steps:

1. From the list of wiki documents, click the document that you want to edit from the list of documents to select it, and then click the Edit Mode icon (the pencil icon). Your document opens in edit mode, and options on the menu bar display.
2. Edit the markdown content in the document or add content to the document. You can format the content using the options on the menu bar.
   A preview of your changes shows up in the pane to the right, the preview pane.
3. Save your changes by clicking **Publish**.

Edit a Wiki Document's Title

Follow these steps:

1. From the list of documents, select the document that you want to edit, and then select **Edit**.
   The Edit pane opens.
2. Edit the title for the document, and then click **Publish**.

The changes to the document title are saved and published.

Reorder a Wiki Document

You can reorder your wiki documents within the document tree, such as moving a document to be a child document of another document or reordering the list of wiki documents.

To reorder using the mouse, from the list of wiki documents, click and hold the document that you want to reorder, and then drag and drop it to the new location in the tree.

To reorder using the keyboard, from the list of wiki documents, pick up the document that you want to reorder by pressing the Spacebar key on your keyboard, navigate to the new position using the arrow keys on your keyboard, and then drop it to the new location in the tree by pressing the Spacebar key again.

Delete a Wiki Document

Deleting a parent document also deletes the child documents within the document tree (recursive delete).

Follow these steps:

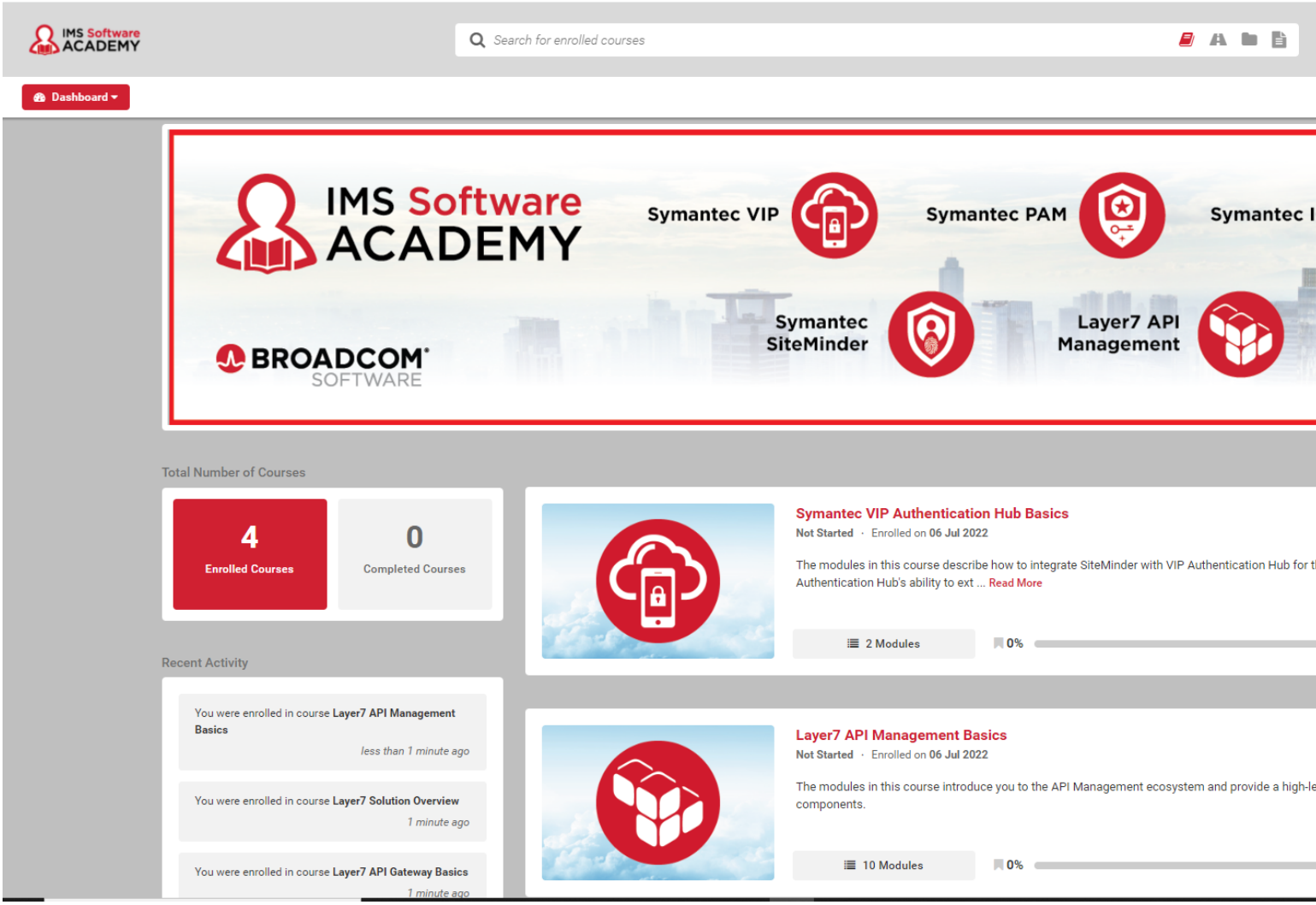
1. From the list of wiki documents, select the document that you want to delete and click **Delete**.
   The Delete confirmation dialog box opens.
2. Confirm the deletion by clicking **Yes**.

Portal Training Videos on IMS Software Academy

Enroll for free training videos on IMS Software Academy.

Broadcom's Identity Management Security (IMS) division offers free education videos to help you get the most out of our IMS products. Visit the IMS Software Academy, create your free account, and get started today.

You can log into IMS Software Academy with your existing Broadcom credentials. If you do not have Broadcom credentials, provide your official email address and start your training.



API Developer Portal Courses

The following API Developer Portal courses are available on IMS Software Academy:

- **Layer7 Solution Overview:** This course includes one module that provides a brief introduction to the Layer7 API Management solution.
- **Layer7 API Developer Portal Basics:** This course help you understand how to publish APIs, enable consumers to discover the available services, and help monitor API performance using API Developer Portal. This course includes the following modules:
  - Introduction to API Portal
  - Introduction to API Portal Administration
  - API Portal Deployment on Kubernetes
  - API Portal Deployment on Kubernetes with High Availability
  - Reporting and Custom Dashboards
  - Layer7 - Managing Multiple Keys Across Environments
  - Layer7 - Managing API and API Key Deployments
  - How do I create and manage Rate Limits and Quotas?
  - Layer7 API Portal - How do I create and manage Organizations?
  - Layer7 - How do I publish an API using the Portal?
  - Layer7 - How do I create an application that consumes an API?
  - Layer7 - How do I create new policy templates for API Publishing? (using bundles)
  - Layer7 - Getting Started with API Hub

Let us know what you would like to learn more about, and we will add it to the list.

Third-Party Software Acknowledgments

Third-party software was used in the creation of API Portal. All third-party software have been used in accordance with the terms and conditions for use, reproduction, and distribution as defined by the applicable license agreements.

Click to download the Third-Party Software Agreements for the API Portal .

Product Accessibility Features

CA Technologies is committed to ensuring that all customers, regardless of ability, can successfully use its products and supporting documentation to accomplish vital business tasks.

Product Enhancements

API Management SaaS offers accessibility enhancements in the following areas:

- Display
- Sound
- Keyboard
- Mouse
- Custom Controls (if any)

Display

To increase visibility on your computer display, you can adjust the following options:

- **Font style, color, and size of items**Defines font color, size, and other visual combinations.
- **Screen resolution**Defines the pixel count to enlarge objects on the screen.
- **Cursor width and blink rate**Defines the cursor width or blink rate, which makes the cursor easier to find or minimize its blinking.
- **Icon size**Defines the size of icons. You can make icons larger for visibility or smaller for increased screen space.
- **High contrast schemes**Defines color combinations. You can select colors that are easier to see.

Sound

Use sound as a visual alternative or to make computer sounds easier to hear or distinguish by adjusting the following options:

- **Volume**Sets the computer sound up or down.
- **Text-to-Speech**Sets the computer's hear command options and text read aloud.
- **Warnings**Defines visual warnings.
- **Notices**Defines the aural or visual cues when accessibility features are turned on or off.
- **Schemes**Associates computer sounds with specific system events.
- **Captions**Displays captions for speech and sounds.

Keyboard

You can make the following keyboard adjustments:

- **Repeat Rate**Defines how quickly a character repeats when a key is struck.
- **Tones**Defines tones when pressing certain keys.
- **Sticky Keys**Defines the modifier key, such as Shift, Ctrl, Alt, or the Windows Logo key, for shortcut key combinations. Sticky keys remain active until another key is pressed.

Mouse

You can use the following options to make your mouse faster and easier to use:

- **Click Speed**Defines how fast to click the mouse button to make a selection.
- **Click Lock**Sets the mouse to highlight or drag without holding down the mouse button.
- **Reverse Action**Sets the reverse function controlled by the left and right mouse keys.
- **Blink Rate**Defines how fast the cursor blinks or if it blinks at all.
- **Pointer Options**Let you complete the following actions:
  - Hide the pointer while typing
  - Show the location of the pointer
  - Set the speed that the pointer moves on the screen
  - Choose the pointer's size and color for increased visibility
  - Move the pointer to a default location in a dialog box

Keyboard Shortcuts

The following table lists the keyboard shortcuts that *API Management SaaS* supports:

Keyboard	Description
Ctrl+X	Cut
Ctrl+C	Copy
Ctrl+K	Find Next
Ctrl+F	Find and Replace
Ctrl+V	Paste
Ctrl+S	Save
Ctrl+Shift+S	Save All
Ctrl+D	Delete Line
Ctrl+Right	Next Word
Ctrl+Down	Scroll Line Down
End	Line End

Documentation Legal Notice

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by Broadcom at any time. This Documentation is proprietary information of Broadcom and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of Broadcom.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all Broadcom copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to Broadcom that all copies and partial copies of the Documentation have been returned to Broadcom or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, BROADCOM PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL BROADCOM BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF BROADCOM IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is Broadcom Inc.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2005-2022 Broadcom. All Rights Reserved. The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

BROADCOM®
SOFTWARE